

PCT/JP 99/02049  
19.04.99

日 本 国 特 許 庁

PATENT OFFICE  
JAPANESE GOVERNMENT

09/673503

REC'D 02 JUL 1999

WIPO PCT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application:

1998年 4月17日

EU

出 願 番 号  
Application Number:

平成10年特許願第124286号

出 願 人  
Applicant (s):

大見 忠弘  
株式会社ウルトラクリーンテクノロジー開発研究所

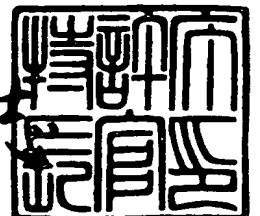
PRIORITY  
DOCUMENT

SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)

1999年 6月17日

特許庁長官  
Commissioner,  
Patent Office

伴佐山 建志



出証番号 出証特平11-3033302

【書類名】	特許願
【整理番号】	OHM0283
【提出日】	平成10年 4月17日
【あて先】	特許庁長官殿
【国際特許分類】	H01L 21/00
【発明の名称】	データ圧縮伸長システム
【請求項の数】	16
【発明者】	
【住所又は居所】	宮城県仙台市青葉区米ヶ袋2の1の17の301
【氏名】	大見 忠弘
【発明者】	
【住所又は居所】	千葉県千葉市美浜区稲毛海岸5丁目5-2-206
【氏名】	小谷 光司
【発明者】	
【住所又は居所】	東京都足立区加平二丁目12番5号
【氏名】	中田 明良
【発明者】	
【住所又は居所】	宮城県仙台市青葉区荒巻字青葉（無番地）東北大学内
【氏名】	今井 誠
【発明者】	
【住所又は居所】	宮城県仙台市青葉区荒巻字青葉（無番地）東北大学内
【氏名】	譽田 正宏
【発明者】	
【住所又は居所】	宮城県仙台市青葉区荒巻字青葉（無番地）東北大学内
【氏名】	森本 達郎
【発明者】	
【住所又は居所】	宮城県仙台市青葉区荒巻字青葉（無番地）東北大学内
【氏名】	米澤 岳美

【発明者】

【住所又は居所】 宮城県仙台市青葉区荒巻字青葉（無番地）東北大学内

【氏名】 野沢 俊之

【発明者】

【住所又は居所】 宮城県仙台市青葉区荒巻字青葉（無番地）東北大学内

【氏名】 中山 貴裕

【発明者】

【住所又は居所】 宮城県仙台市青葉区荒巻字青葉（無番地）東北大学内

【氏名】 藤林 正典

【発明者】

【住所又は居所】 東京都文京区本郷4丁目1番4号 株式会社ウルトラクリーンテクノロジー開発研究所内

【氏名】 新田 雄久

【特許出願人】

【識別番号】 000205041

【氏名又は名称】 大見 忠弘

【特許出願人】

【識別番号】 596089517

【氏名又は名称】 株式会社ウルトラクリーンテクノロジー開発研究所

【代表者】 新田 雄久

【代理人】

【識別番号】 100088096

【弁理士】

【氏名又は名称】 福森 久夫

【電話番号】 03-3261-0690

【手数料の表示】

【予納台帳番号】 007467

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9712127

【包括委任状番号】 9712234

【書類名】 明細書

【発明の名称】 データ圧縮伸長システム

【特許請求の範囲】

【請求項 1】 少なくとも 1 つ以上のデータからなるデータ列を第 1 のベクトルとし、少なくとも 1 つ以上の組からなる前記第 1 のベクトル群からなる静止画コードブックを格納しておく記憶装置を有し、

外部から入力された一枚または複数の画像データに対し、前記画像データを前記静止画コードブックにより、データ圧縮、伸長を行った結果を静止画デコード用データ列群とし、前記静止画デコード用データ列群の個々の静止画デコード用データ列の個々のデータに対応するアドレスの前記静止画コードブックのデータを抜きだし、所望の方法に従って、前記静止画コードブックのデータを 1 つの画面内でブロック毎にずらして表示装置に表示させることを特徴とするデータ圧縮システム。

【請求項 2】 少なくとも 1 つ以上のデータからなるデータ列を第 1 のベクトルとし、少なくとも 1 つ以上の組からなる前記第 1 のベクトル群からなる静止画コードブックを格納しておく記憶装置を有し、

外部から入力された複数の画像に対し、各々の画像から所望の複数の画素からなるブロックを抜き出し、前記ブロックの画素が持つ値からなるデータ列を入力ベクトルとし、1 つまたは複数の前記入力ベクトルのうち任意のベクトルの個々のデータと前記静止画コードブックを構成するベクトル群の個々の第 1 のベクトルのデータ列のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さい第 1 のベクトルが格納されている前記記憶装置のアドレスの特定を行い、前記入力ベクトルに対応する入力画像のブロックアドレスが持つ静止画コード番号データとして、特定した前記記憶装置のアドレス番号を割り当て、1 枚または複数の入力画像に対して各々静止画コード番号データ列を生成するデータ圧縮装置であって、

前記ブロックを画素が一次元に配列したラインブロックとすることを特徴とするデータ圧縮システム。

【請求項 3】 少なくとも 1 つ以上のデータからなるデータ列を第 1 のベク

トルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなる静止画コードブックを格納しておく第1の記憶装置と、少なくとも1つ以上のデータからなるデータ列を第2のベクトルとし、少なくとも1つ以上の組からなる前記第2のベクトル群からなる時間軸コードブックを格納しておく第2の記憶装置を有し、

外部から入力された複数の画像に対し、各々の画像から所望の複数の画素からなるブロックを抜き出し、前記ブロックの画素が持つ値からなるデータ列を入力ベクトルとし、1つまたは複数の前記入力ベクトルのうち任意のベクトルの個々のデータと前記静止画コードブックを構成するベクトル群の個々の第1のベクトルのデータ列のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さい第1のベクトルが格納されている前記第1の記憶装置のアドレスの特定を行い、前記入力ベクトルに対応する入力画像のブロックアドレスが持つ静止画コード番号データとして、特定した前記記憶装置のアドレス番号を割り当て、1枚または複数の入力画像に対して各々静止画コード番号データ列を生成し、

前記静止画コード番号データ列群の個々の静止画コード番号データ列の同じアドレスに存在するデータを抜きだし、入力された画像の順序に配列された静止画コード番号フレームベクトルを生成し、前記静止画コード番号フレームベクトル群の個々の静止画コード番号フレームベクトルのデータと前記時間軸コードブックを構成する第2のベクトル群の個々のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さい第2のベクトルが格納されている前記第2の記憶装置のアドレスの特定を行い、前記静止画コード番号フレームベクトルに対応するアドレスが持つ時間軸コード番号データとして、前記第2の記憶装置のアドレス番号を割り当て、

時間軸コード番号データ列を生成することを特徴とするデータ圧縮システム。

【請求項4】 少なくとも1つ以上のデータからなるデータ列を第1のベクトルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなる静止画コードブックを格納しておく第1の記憶装置と、少なくとも1つ以上のデータからなるデータ列を第2のベクトルとし、少なくとも1つ以上の組からなる前記

第2のベクトル群からなる時間軸コードブックを格納しておく第2の記憶装置を有し、

外部から入力された複数の画像に対し、各々の画像から所望の複数の画素からなるブロックを抜き出し、前記ブロックの画素が持つ値からなるデータ列を入力ベクトルとし、1つまたは複数の前記入力ベクトルのうち任意のベクトルの個々のデータと前記静止画コードブックを構成するベクトル群の個々の第1のベクトルのデータ列のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さい第1のベクトルが格納されている前記第1の記憶装置のアドレスの特定を行い、前記入力ベクトルに対応する入力画像のブロックアドレスが持つ静止画コード番号データとして、特定した前記第1の記憶装置のアドレス番号を割り当て、1枚または複数の入力画像に対して各々静止画コード番号データ列を生成し、

前記第1の静止画コード番号データ列を基準静止画コード番号データ列とし、前記基準静止画コード番号データ列と前記静止画コード番号データ列の同じアドレスに存在する個々のデータの差分をとり、所望の順序に配列された静止画コード番号フレーム差分ベクトルを生成し、前記静止画コード番号フレーム差分ベクトル群の個々の静止画コード番号フレーム差分ベクトルのデータと前記時間軸コードブックを構成する第2のベクトル群の個々のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さい第2のベクトルが格納されている前記第2の記憶装置のアドレスの特定を行い、前記静止画コード番号フレームベクトルに対応するアドレスが持つ時間軸コード番号データとして、前記第2の記憶装置のアドレス番号を割り当て、

時間軸コード番号データ列を生成することを特徴とするデータ圧縮システム。

【請求項5】 少なくとも1つ以上のデータからなるデータ列を第1のベクトルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなる静止画コードブックを格納しておく第1の記憶装置と、少なくとも1つデータの組からなるデータ列を第2のベクトルとし、少なくとも1つ以上の組からなる前記第2のベクトル群からなる時間軸コードブックを格納しておく第2の記憶装置を有し、

外部から入力された複数の画像に対し、各々の画像から所望の複数の画素からなるブロックを抜き出し、前記ブロックの画素が持つ値からなるデータ列を入力ベクトルとし、1つまたは複数の前記入力ベクトルのうち任意のベクトルの個々のデータと前記静止画コードブックを構成するベクトル群の個々の第1のベクトルのデータ列のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さい第1のベクトルが格納されている前記第1の記憶装置のアドレスの特定を行い、前記入力ベクトルに対応する入力画像のブロックアドレスが持つ静止画コード番号データとして、特定した前記第1の記憶装置のアドレス番号を割り当て、1枚または複数の入力画像に対して各々静止画コード番号データ列を生成し、

所望の前記第1の静止画コード番号データ列と所望の前記第2の静止画コード番号データ列の同じアドレスに存在する個々のデータの差分をとり、所望の順序に配列された静止画コード番号フレーム差分ベクトルを生成し、前記静止画コード番号フレーム差分ベクトル群の個々の静止画コード番号フレーム差分ベクトルのデータと前記時間軸コードブックを構成する第2のベクトル群の個々のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さい第2のベクトルが格納されている前記第2の記憶装置のアドレスの特定を行い、前記静止画コード番号フレームベクトルに対応するアドレスが持つ時間軸コード番号データとして、前記記憶装置のアドレス番号を割り当て、時間軸コード番号データ列を生成することを特徴とするデータ圧縮システム。

【請求項6】 少なくとも1つ以上のデータからなるデータ列を第1のベクトルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなる静止画コードブックを格納しておく第1の記憶装置と、少なくとも1つ以上の組からなるデータ列を第2のベクトルとし、少なくとも1つ以上の組からなる前記第2のベクトル群からなる時間軸コードブックを格納しておく第2の記憶装置を有し、

外部から入力された複数の画像データに対し、前記画像データを前記静止画コードブック並びに前記時間軸コードブックによりデータ圧縮、伸長を行った結果を静止画デコード用データ列群とし、前記静止画デコード用データ列群の個々の



静止画デコード用データ列の個々のデータに対応するアドレスの前記静止画コードブックのデータを抜きだし、所望の方法に従って、前記静止画コードブックのデータをフレーム間でブロック毎にずらして表示装置に表示させることを特徴とする請求項3～5のいずれか1項に記載のデータ圧縮システム。

【請求項7】 少なくとも1つ以上のデータからなるデータ列を第1のベクトルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなる静止画コードブックを格納しておく第1の記憶装置を有し、

外部から入力された複数の入力画像データに対し、前記入力画像データを前記静止画コードブックによりデータ圧縮を行った結果を静止画圧縮データ列とし、前記静止画圧縮データ列を格納しておく第2の記憶装置を有し、

前記第2の記憶装置に格納されている静止画圧縮データ列と、そこに格納されている静止画圧縮データ列とは異なる静止画圧縮データ列の各々のデータに対応するアドレスにあるデータ同士で差分絶対値を取る装置を有し、

前記差分絶対値を取る装置からの出力が所望の値よりも大きな値となった場合にのみデータを転送することを特徴とするデータ圧縮システム。

【請求項8】 少なくとも1つ以上のデータからなるデータ列を第1のベクトルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなる静止画コードブックを格納しておく第1の記憶媒体を有し、

外部から入力された複数の入力画像データに対し、前記入力画像データを前記静止画コードブックによりデータ圧縮を行った結果を静止画圧縮データ列とし、前記静止画圧縮データ列を格納しておく第2の記憶媒体を有し、

前記第2の記憶媒体に格納されている静止画圧縮データ列と、そこに格納されている静止画圧縮データ列とは異なる静止画圧縮データ列の各々のデータに対応するアドレスにあるデータ同士で差分絶対値を取り、

前記差分絶対値が所望の値よりも大きな値となった場合のみデータを転送することを特徴とするデータ圧縮システム。

【請求項9】 1つ以上のデータからなるデータ列をベクトルとし、前記ベクトルが1つ以上組み合わせさせたものをコードブックとし、仮想の2次元平面上で、前記ベクトル群を複数格納しておく記憶装置に格納してあるベクトル群の個

々のデータと、ある入力データ列の個々のデータとの差分をとり、前記差分結果を絶対値表現したそれぞれの値を加算した結果の最も小さいベクトルが格納されている前記記憶装置のアドレスの特定を行い、前記特定されたアドレスのベクトルを含む、ある範囲のアドレスを更新先アドレスとし、

前記入力データ列と前記更新先アドレスで指定された前記記憶装置内の1つまたは複数のベクトルを入力とし、前記入力データ列の個々のデータと前記更新先アドレスで指定された前記記憶装置内のベクトルの個々のデータとの差分をとり、前記差分結果に所望の係数を乗算し、前記更新先アドレスで指定された前記記憶装置内のベクトルの個々のデータに加えたものを更新データ列として出力し、前記更新先アドレスで指定された前記記憶装置内のコードブックを更新コードブックに変更する操作を複数回繰り返すコードブック変換方法において、

コードブックの初期の値として、とり得る値の最小値から最大値まで連続的に変化することを特徴とする請求項1～8のいずれか1項に記載のデータ圧縮システム。

【請求項10】 1つ以上のデータからなるデータ列をベクトルとし、前記ベクトルが1つ以上組み合わさったものをコードブックとする。

仮想の2次元平面上で、前記ベクトル群を複数格納しておく記憶装置に格納してあるベクトル群の個々のデータと、ある入力データ列の個々のデータとの差分をとり、前記差分結果を絶対値表現したそれぞれの値を加算した結果の最も小さいベクトルが格納されている前記記憶装置のアドレスの特定を行い、前記特定されたアドレスのベクトルを含む、ある範囲のアドレスを更新先アドレスとし、

前記入力データ列と前記更新先アドレスで指定された前記記憶装置内の1つまたは複数のベクトルを入力とし、前記入力データ列の個々のデータと前記更新先アドレスで指定された前記記憶装置内のベクトルの個々のデータとの差分をとり、前記差分結果に所望の係数を乗算し、前記更新先アドレスで指定された前記記憶装置内のベクトルの個々のデータに加えたものを更新データ列として出力し、前記更新先アドレスで指定された前記記憶装置内のコードブックを更新コードブックに変更する操作を複数回繰り返すコードブック変換方法において、

前記範囲を前記2次元平面上で1次元的に適用し、かつ前記更新回数の増加と

ともに減少させることを特徴とする請求項 1～8 のいずれか 1 項に記載のデータ圧縮システム。

【請求項 11】 前記係数の初期値を 0.7 から 1 とし、前記更新回数の増加とともに減少させることを特徴とする請求項 9 または請求項 10 のいずれかに記載のデータ圧縮システム。

【請求項 12】 請求項 9～11 に記載のデータ圧縮システムによって作成された複数のコードブックを合成して新しいコードブックとすることを特徴とするデータ圧縮システム。

【請求項 13】 記憶された複数のテンプレートベクトルデータの中から、画像データから構成された入力ベクトルデータに類似したテンプレートベクトルを検索し、検索されたテンプレートベクトルデータの値、又はその番地、またはテンプレートベクトルに付加されたインデックス番号を出力する検索手段を有するデータ圧縮システムであって、前記検索を行う際に、前記入力ベクトルデータ及びテンプレートベクトルデータの特徴量を比較し、その結果に基づいて、検索のために必要な演算を減少させることを特徴とする請求項 1～8 のいずれか 1 項に記載のデータ圧縮装置。

【請求項 14】 前記特徴量は、ベクトルデータの一部の要素の値を画素の明暗が反転するように操作した後のベクトルデータの要素の総和若しくは平均値又は直流成分を特徴とする請求項 13 に記載のデータ圧縮システム。

【請求項 15】 前記検索手段は、テンプレートベクトルデータの個々の要素と前記入力ベクトルデータの対応する要素との間で差分をとり、その絶対値の総和であるマンハッタン距離が最小となるテンプレートベクトルデータを検出するものであって、任意のテンプレートベクトルデータについて、入力ベクトルデータとの間で前記マンハッタン距離  $M$  を演算し、他のテンプレートベクトルデータの個々の要素の総和と入力ベクトルデータの個々の要素の総和との差分の絶対値  $D$  とを比較して、 $M < D$  の関係を有する場合は、該他のテンプレートベクトルデータについては、入力ベクトルデータとの間でマンハッタン距離の演算を省略することを特徴とする請求項 13 または 14 に記載のデータ圧縮システム。

【請求項 16】 少なくとも 1 つ以上のデータからなるデータ列を第 1 のベ

クトルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなるコードブックを格納しておく記憶装置を有し、

前記記憶装置にに対し、前記コードブックを書き込む登録装置を有し、

前記記憶装置に格納されている前記コードブックに対し、検索・並び替え・読み出し・書き替えを行う管理装置を有し、

前記コードブックを生成あるいは前記記憶装置に格納されている前記コードブックを変換する生成装置を有し、

外部からのコードブック送信命令に従って、前記記憶装置に格納されている1つ以上のコードブックもしくは前記生成装置により生成あるいは変換された1つ以上のコードブックを送信することを特徴としたデータ圧縮伸長システム。

#### 【発明の詳細な説明】

##### 【0001】

##### 【発明の属する技術分野】

本発明はデータ圧縮システム及びデータ圧縮伸長システムに関する。

##### 【0002】

##### 【従来の技術】

従来、データ圧縮システムとしては、伸長が非常に簡単にできるデータ圧縮アルゴリズムの1つとしてベクトル量子化という手法が良く知られている。このアルゴリズムは古くから信号処理の分野で知られており、特に画像や音声信号のデータ圧縮やパターン認識に応用されてきた。ベクトル量子化は、2次元配列の画像データからある大きさ（例えば4×4画素）のブロックを取り出し、その中で最も似通ったパターンを見つけ出し、そのパターンの番号を当てはめるというデータ圧縮を行う。受信者はコードブックの中から番号に対応するパターンを取り出すだけで画像を再現することができる。従って、コードブックさえ受け取っていれば特に特殊な演算は必要としないため非常に簡単なハードウェアで画像を再生することが可能となる。

##### 【0003】

画像や音声信号のデータ圧縮をベクトル量子化で行う際に、高い圧縮率を保持したままいかに高画質の再生画像を得るのか、またベクトル量子化を利用する上

で必ず必要となるコードブックをいかに性能良いものを作成するかが課題となっている。高い圧縮率を実現するために、画像データから取り出すブロックの大きさを大きくしたり、コードブックを構成するブロックの個数を少なくするといった努力がなされている。また、コードブックの最適化の手法としてはKohonenの自己組織化マップの手法などいくつか知られている。

## 【0004】

しかし、実際に動画像を高い圧縮率で圧縮・伸長したり、コードブックの最適化技術には以下のような問題点がある。

## 【0005】

画像を高い圧縮率でベクトル量子化によりデータ圧縮を行うと、どうしても再生画像の画質が落ちてしまうという問題がある。また、再生画像の画質を高めようとすると、ベクトル量子化以外の処理が入るため、結果的に圧縮率は高くないといった問題点があった。画像データの圧縮に用いられるベクトル量子化技術は、静止画に対してベクトル量子化を行い、動画像を転送する場合にはベクトル量子化技術以外の技術を組み合わせて圧縮を行い、実現されてきていた。しかしながら、圧縮率が結果的に高くない割には、再生画像の画質が悪くなっていた。

## 【0006】

また、コードブックの最適化技術に関しては、最適化に使用したデータのみ有用なコードブックとなってしまうことである。

## 【0007】

例えば、顔の画像で最適化されたコードブックは、最適化に用いた画像に対して最良のコードブックとなる。そのコードブックを他の画像に対して用いた時、人の顔という分類に含まれる画像に対しては比較的画質の良い再生画像が得られるが、風景や文字といった異なる分類の画像に対しては画質の悪い再生画像となってしまう。つまり、コードブックのパターンが画像によって全く異なっているために、汎用性の低いコードブックになってしまう。

## 【0008】

【発明が解決しようとする課題】

本発明は、動画像や音声信号のデータ圧縮を高圧縮率で実現し、圧縮データをもとに質の高いデータを再生しすることを目的とする。

【0009】

【課題を解決するための手段】

本発明のデータ圧縮システムは、少なくとも1つ以上のデータからなるデータ列を第1のベクトルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなる静止画コードブックを格納しておく記憶装置において、外部から入力された一枚または複数の画像データに対し、前記画像データを前記静止画コードブックにより、データ圧縮、伸長を行った結果を静止画デコード用データ列群とし、前記静止画デコード用データ列群の個々の静止画デコード用データ列の個々のデータに対応するアドレスの前記静止画コードブックのデータを抜きだし、所望の方法に従って、前記静止画コードブックのデータを1つの画面内でブロック毎にずらして表示装置に表示させることを特徴とする。

【0010】

本発明のデータ圧縮システムは、少なくとも1つ以上のデータからなるデータ列を第1のベクトルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなる静止画コードブックを格納しておく記憶装置を有し、

外部から入力された複数の画像に対し、各々の画像から所望の複数の画素からなるブロックを抜き出し、前記ブロックの画素が持つ値からなるデータ列を入力ベクトルとし、1つまたは複数の前記入力ベクトルのうち任意のベクトルの個々のデータと前記静止画コードブックを構成するベクトル群の個々の第1のベクトルのデータ列のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さい第1のベクトルが格納されている前記記憶装置のアドレスの特定を行い、前記入力ベクトルに対応する入力画像のブロックアドレスが持つ静止画コード番号データとして、特定した前記記憶装置のアドレス番号を割り当て、1枚または複数の入力画像に対して各々静止画コード番号データ列を生成するデータ圧縮装置であって、

前記ブロックを画素が一次元に配列したラインブロックとすることを特徴とする。

## 【0011】

本発明のデータ圧縮システムは、少なくとも1つ以上のデータからなるデータ列を第1のベクトルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなる静止画コードブックを格納しておく第1の記憶装置と、少なくとも1つ以上のデータからなるデータ列を第2のベクトルとし、少なくとも1つ以上の組からなる前記第2のベクトル群からなる時間軸コードブックを格納しておく第2の記憶装置を有し、

外部から入力された複数の画像に対し、各々の画像から所望の複数の画素からなるブロックを抜き出し、前記ブロックの画素が持つ値からなるデータ列を入力ベクトルとし、1つまたは複数の前記入力ベクトルのうち任意のベクトルの個々のデータと前記静止画コードブックを構成するベクトル群の個々の第1のベクトルのデータ列のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さい第1のベクトルが格納されている前記第1の記憶装置のアドレスの特定を行い、前記入力ベクトルに対応する入力画像のブロックアドレスが持つ静止画コード番号データとして、特定した前記記憶装置のアドレス番号を割り当て、1枚または複数の入力画像に対して各々静止画コード番号データ列を生成し、

前記静止画コード番号データ列群の個々の静止画コード番号データ列の同じアドレスに存在するデータを抜きだし、入力された画像の順序に配列された静止画コード番号フレームベクトルを生成し、前記静止画コード番号フレームベクトル群の個々の静止画コード番号フレームベクトルのデータと前記時間軸コードブックを構成する第2のベクトル群の個々のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さい第2のベクトルが格納されている前記第2の記憶装置のアドレスの特定を行い、前記静止画コード番号フレームベクトルに対応するアドレスが持つ時間軸コード番号データとして、前記第2の記憶装置のアドレス番号を割り当て、時間軸コード番号データ列を生成することを特徴とする。

## 【0012】

本発明のデータ圧縮システムは、少なくとも1つ以上のデータからなるデータ

列を第1のベクトルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなる静止画コードブックを格納しておく第1の記憶装置と、少なくとも1つ以上のデータからなるデータ列を第2のベクトルとし、少なくとも1つ以上の組からなる前記第2のベクトル群からなる時間軸コードブックを格納しておく第2の記憶装置を有し、

外部から入力された複数の画像に対し、各々の画像から所望の複数の画素からなるブロックを抜き出し、前記ブロックの画素が持つ値からなるデータ列を入力ベクトルとし、1つまたは複数の前記入力ベクトルのうち任意のベクトルの個々のデータと前記静止画コードブックを構成するベクトル群の個々の第1のベクトルのデータ列のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さい第1のベクトルが格納されている前記第1の記憶装置のアドレスの特定を行い、前記入力ベクトルに対応する入力画像のブロックアドレスが持つ静止画コード番号データとして、特定した前記第1の記憶装置のアドレス番号を割り当て、1枚または複数の入力画像に対して各々静止画コード番号データ列を生成し、前記第1の静止画コード番号データ列を基準静止画コード番号データ列とし、前記基準静止画コード番号データ列と前記静止画コード番号データ列の同じアドレスに存在する個々のデータの差分をとり、所望の順序に配列された静止画コード番号フレーム差分ベクトルを生成し、前記静止画コード番号フレーム差分ベクトル群の個々の静止画コード番号フレーム差分ベクトルのデータと前記時間軸コードブックを構成する第2のベクトル群の個々のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さい第2のベクトルが格納されている前記第2の記憶装置のアドレスの特定を行い、前記静止画コード番号フレームベクトルに対応するアドレスが持つ時間軸コード番号データとして、前記第2の記憶装置のアドレス番号を割り当て、時間軸コード番号データ列を生成することを特徴とする。

【0013】

本発明のデータ圧縮システムは、少なくとも1つ以上のデータからなるデータ列を第1のベクトルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなる静止画コードブックを格納しておく第1の記憶装置と、少なくとも1



つデータの組からならデータ列を第2のベクトルとし、少なくとも1つ以上の組からなる前記第2のベクトル群からなる時間軸コードブックを格納しておく第2の記憶装置を有し、

外部から入力された複数の画像に対し、各々の画像から所望の複数の画素からなるブロックを抜き出し、前記ブロックの画素が持つ値からなるデータ列を入力ベクトルとし、1つまたは複数の前記入力ベクトルのうち任意のベクトルの個々のデータと前記静止画コードブックを構成するベクトル群の個々の第1のベクトルのデータ列のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さい第1のベクトルが格納されている前記第1の記憶装置のアドレスの特定を行い、前記入力ベクトルに対応する入力画像のブロックアドレスが持つ静止画コード番号データとして、特定した前記第1の記憶装置のアドレス番号を割り当て、1枚または複数の入力画像に対して各々静止画コード番号データ列を生成し、

所望の前記第1の静止画コード番号データ列と所望の前記第2の静止画コード番号データ列の同じアドレスに存在する個々のデータの差分をとり、所望の順序に配列された静止画コード番号フレーム差分ベクトルを生成し、前記静止画コード番号フレーム差分ベクトル群の個々の静止画コード番号フレーム差分ベクトルのデータと前記時間軸コードブックを構成する第2のベクトル群の個々のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さい第2のベクトルが格納されている前記第2の記憶装置のアドレスの特定を行い、前記静止画コード番号フレームベクトルに対応するアドレスが持つ時間軸コード番号データとして、前記記憶装置のアドレス番号を割り当て、時間軸コード番号データ列を生成することを特徴とする。

#### 【0014】

本発明のデータ圧縮システムは、少なくとも1つ以上のデータからなるデータ列を第1のベクトルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなる静止画コードブックを格納しておく第1の記憶装置を有し、

外部から入力された複数の入力画像データに対し、前記入力画像データを前記静止画コードブックによりデータ圧縮を行った結果を静止画圧縮データ列とし、

前記静止画圧縮データ列を格納しておく第2の記憶装置を有し、

前記第2の記憶装置に格納されている静止画圧縮データ列と、そこに格納されている静止画圧縮データ列とは異なる静止画圧縮データ列の各々のデータに対応するアドレスにあるデータ同士で差分絶対値を取る装置を有し、

前記差分絶対値を取る装置からの出力が所望の値よりも大きな値となった場合にのみデータを転送することを特徴とする。

#### 【0015】

本発明のデータ圧縮システムは、少なくとも1つ以上のデータからなるデータ列を第1のベクトルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなる静止画コードブックを格納しておく第1の記憶媒体を有し、

外部から入力された複数の入力画像データに対し、前記入力画像データを前記静止画コードブックによりデータ圧縮を行った結果を静止画圧縮データ列とし、前記静止画圧縮データ列を格納しておく第2の記憶媒体を有し、

前記第2の記憶媒体に格納されている静止画圧縮データ列と、そこに格納されている静止画圧縮データ列とは異なる静止画圧縮データ列の各々のデータに対応するアドレスにあるデータ同士で差分絶対値を取り、

前記差分絶対値が所望の値よりも大きな値となった場合のみデータを転送することを特徴とする。

#### 【0016】

本発明のデータ圧縮伸長システムは、少なくとも1つ以上のデータからなるデータ列を第1のベクトルとし、少なくとも1つ以上の組からなる前記第1のベクトル群からなるコードブックを格納しておく記憶装置を有し、

前記記憶装置にに対し、前記コードブックを書き込む登録装置を有し、

前記記憶装置に格納されている前記コードブックに対し、検索・並び替え・読み出し・書き替えを行う管理装置を有し、  
前記コードブックを生成あるいは前記記憶装置に格納されている前記コードブックを変換する生成装置を有し、

外部からのコードブック送信命令に従って、前記記憶装置に格納されている1つ以上のコードブックもしくは前記生成装置により生成あるいは変換された1つ

以上のコードブックを送信することを特徴とする。

【0017】

【実施例】

以下に実施例を挙げて本発明をより詳細に説明する。

【0018】

(実施例1)

本実施例では、図1の写真に示すベクトル量子化PCIバスボードを用いて、動画像をベクトル量子化により圧縮を行い、圧縮を行った結果から動画像の再生を行うシステムを実現している。以下に、その詳細な説明を行う。

【0019】

図1に示すベクトル量子化PCIバスボードには8ビット16次元のコードブックベクトルが256個登録可能なデジタルベクトル量子化チップが8個実装されており、合わせて2048個のコードブックベクトルに対するベクトル量子化が実行可能となっている。これらのチップはFPGAにより構成されたPCIバスインターフェースを介してPCと接続され、ベクトル量子化のデモンストレーションが可能となっている。

【0020】

図2にここで構築したベクトル量子化システム全体のデータの流れを、このボードを含めて示している。ベクトル量子化はメモリ上に保存した動画像データを使用して行った。動作の流れは以下のとおりである。

【0021】

まず予め作成しておいたコードブックを構成している全てのコードブックベクトルをメモリからベクトル量子化チップ上に転送しておく。次に、メモリに記憶されている動画像データを順次ベクトル量子化ボードに送りベクトル量子化を実行する。ベクトル量子化の結果得られたコードは一旦メモリに入り、このコードをもとにそれに対応するコードブックベクトルをメモリから読み出し、デコードを実行する。デコードされたデータはディスプレイに送られ、動画像として表示される。図3にこのようにして得られた動画像の1フレームを示す。

【0022】

図4は640×480画素の一枚の画像を処理する際に、それぞれの処理にかかる所用時間を測定した結果である。ベクトル量子化に約50 msec、デコードに約160 msec、表示に約60 msecを要している。デコードにこれだけの時間がかかる理由は、これまで述べてきたベクトル量子化の手法だと一つのベクトルが画像上の4×4画素の正方形のブロックに対応しており、コードで指定されたコードブックベクトルをメモリから取り出しディスプレイに表示できる形式にする際にアドレス変換が必要なためである。この様な場合、一般的にアドレス変換用の専用チップを用意しその高速化を図るのだがここではもっと簡単に高速化を図る手法について述べる。

### 【0023】

図5にはベクトルとして画像上の走査線に沿ったラインブロックをベクトルデータとして用いた場合（ラインブロック方式）について説明している。正方形のブロックの場合に必要なアドレス変換操作はブロックをライン上に取ることで回避できる。また、この場合のコードブックもKohonenの自己組織化マップの手法を用いれば容易に得ることができる。図4にはラインブロック方式を取った場合の各処理時間も示している。ベクトル量子化および表示走査にかかる時間は変化はないが、デコード時間は160 msecから大幅に短縮できていることが分かる。

### 【0024】

また、ラインブロック方式を取った場合、順番に整然とパターンを貼り込んで画像を表示するとラインの両端のパターンの大きな不連続点がどうしても目立ってしまう。そこで図6に示すようにパターンを貼り込む位置を少しずつずらす工夫を取った。図7はこのようにして再生した画像を示している。画像の両端に毛羽立ったような部分は見えるが概ね良好な画像が得られることが分かった。この画像の両端部分の乱れは表示するときにマスクをかけることで容易に除去することができると考えられる。この工夫によりここでのシステムにおいても640×480画素の動画像に対して毎秒12フレーム程度のベクトル量子化動作は実現できていることが確認できた。表示部分の処理時間の短縮は表示プログラムの改善によって実現可能である。また現在のシステムではベクトル量子化処理の時間

はその6割程度が入力ベクトルをベクトル量子化ボードに転送するためのアドレス変換および転送時間で占められており、実際にベクトル量子化チップが動作しているのは25 msec程度である。したがって転送バス幅を拡張しCPUの処理能力を高めることにより処理速度を短縮することは可能である。これらの改善を実行することにより毎秒30フレームのベクトル量子化動作は実現可能であると考ええる。

#### 【0025】

ここでは、使用したコードブックを構成しているコードブックベクトルの個数を2048個とし、コードブックベクトルを構成しているデータの個数を16個としているが、これは1つの具体例でありここで述べた数値に制限されるものでないことは言うまでもない。また、ここで構成したシステムで用いた画像サイズ、並びにベクトル量子化を行う際のブロックサイズについても、1つの具体例でありここで述べた数値に制限されるものでないことは言うまでもない。

#### 【0026】

##### (実施例2)

ここでは、図8に示すように640×480画素からなり、1画素当たりの輝度値を8bit表現された動画像に対して圧縮伸長を行う方法について述べる。ここでは、16フレーム分の画像に対して圧縮を行っている。

#### 【0027】

あらかじめ準備しておいた1枚の画像に対して、ベクトル量子化を行うための静止画用コードブックと呼ばれるデータベースを用意しておく。ここでは、この静止画用コードブックは16個のデータからなるデータ列を1つのベクトルとし、そのベクトルを複数組み合わせたものである。ここでは、2048ベクトル分を組み合わせ構成している。

#### 【0028】

まず、第1フレーム目の画像を前記静止画用コードブックを用いて、ベクトル量子化を行う。ここでは、第1フレーム目の画像をX方向に4画素、Y方向に4画素ずつ区切ったブロック領域をそれぞれ入力ベクトルと定義し、その一つの入力ベクトルを構成している個々のデータと、静止画用コードブックを構成してい

る各ベクトルの個々のデータとの差分絶対値をもとめ、そのそれぞれの結果を加算し、その加算結果を比較することにより、入力ベクトルにもっとも似通ったベクトルのコード番号を特定する。

#### 【0029】

この操作を前記入力ベクトル全てに対して行い、第1フレーム目の画像を4×4画素に分割したブロック領域を静止画用コードブックのベクトルのコード番号で置き換える。

#### 【0030】

次に、第2のフレーム目の画像から第16のフレーム目の画像まで計15枚の画像に対して、上記の第1のフレーム目と同じ操作を行い、それぞれの画像を4×4画素に分割したブロック領域を静止画用コードブックのベクトルのコード番号で置き換える。

#### 【0031】

静止画用コードブックのコード番号で表現された各フレームの画像に対して、同じアドレスが示すブロックのコード番号をフレーム画像順に並び変える。つまり、静止画ベクトル量子化を行った16フレーム分の画像の場合、1フレーム目の画像の指定されたアドレスのブロックのコード番号が新しく作られるデータ列の0番地の位置におかれ、2フレーム目の画像の1フレーム目の画像と同じアドレスのブロックのコード番号がデータ列の1番地の位置に置かれる。このように、16フレーム分のコード番号に対して並び代えが行われ、最後の16フレーム目の画像の同じアドレスのブロックのコード番号は、データ列の15番地に置かれることになる。ここで新しく生成された16個のコード番号列を含む集合をベクトルとする。

#### 【0032】

この操作を静止画ベクトル量子化を行った画像のすべての画面に対して行うと、640×480画素の画像の場合でかつ一つのブロック領域を4×4画素とした場合、 $160 \times 120 = 19200$ 個のブロックが生成されるので、19200個のベクトルが新しく生成される。

#### 【0033】

次に、ここでは16個のデータで構成された2048個のベクトルからなる時間軸コードブックを用意しておき、先に生成された19200個のベクトルの内ある一つのベクトルの個々のデータと時間軸コードブックを構成するそれぞれのベクトルの個々のデータとの差分絶対値を演算し、各データにおける差分絶対値を加算し、その加算結果がもっとも小さい値を示したベクトルの時間軸コードブック内のアドレスを特定する。

## 【0034】

同様の操作を19200個すべてのベクトルに対して行い、データ列で表現されていたベクトル列を時間軸コードブックの時間軸コード番号列で表現する。されているために、時間軸コード番号列には19200個のデータがあるが、上記の処理により順番に配列されたデータ列は307200個のデータから構成される。

## 【0035】

次に、配列されたデータ列をここでは16個おきにデータを16分割し、19200個のデータからなる静止画デコード用データ列を生成する。

## 【0036】

次に、16個に分割された静止画デコード用データ列の個々のコード番号に対応するコードをあらかじめ転送されている静止画用コードブックから参照し、参照してきた画像データをはめこみもとの画像を再生する。

## 【0037】

これにより、もとの16フレーム分の画像を再生することが出来る。

## 【0038】

ここでは、1つのベクトルを構成するデータを16個だとし、静止画用コードブックが2048ベクトルで構成されているとし、時間軸コードブックが2048ベクトルで構成されているとしたが、これは実施例を述べるにあたり分かりやすくするためであり、ここで述べた数値に固定されるべきでないのは言うまでもない。また、ここでは16フレーム分の画像を時間軸ベクトル量子化により圧縮を行うことにしたが、圧縮を行うフレームの枚数を16フレームに限定する必要はなく、必要な枚数のフレームをベクトル量子化により圧縮を行ってよいことは

言うまでもない。また、ここでは動画像を用いた実施例を示しているが、動画像に限らずに音声信号を対象にしても本発明の効果になんら変わりはない。

#### 【0039】

##### (実施例3)

ここでは、図9に示すように640×480画素からなり、1画素当たりの輝度値を8bit表現された動画像に対して圧縮伸長を行う方法について述べる。ここでは、17フレーム分の画像に対して圧縮を行っている。

#### 【0040】

あらかじめ準備しておいた1枚の画像に対して、ベクトル量子化を行うための静止画用コードブックと呼ばれるデータベースを用意しておく。ここでは、この静止画用コードブックは16個のデータからなるデータ列を1つのベクトルとし、そのベクトルを複数組み合わせたものである。ここでは、2048ベクトル分を組み合わせ構成している。

#### 【0041】

まず、第1フレーム目の画像を前記静止画用コードブックを用いて、ベクトル量子化を行う。ここでは、第1フレーム目の画像をX方向に4画素、Y方向に4画素ずつ区切ったブロック領域をそれぞれ入力ベクトルと定義し、その一つの入力ベクトルを構成している個々のデータと、静止画用コードブックを構成している各ベクトルの個々のデータとの差分絶対値をもとめ、そのそれぞれの結果を加算し、その加算結果を比較することにより、入力ベクトルにもっとも似通ったベクトルのコード番号を特定する。

#### 【0042】

この操作を前記入力ベクトル全てに対して行い、第1フレーム目の画像を4×4画素に分割したブロック領域を静止画用コードブックのベクトルのコード番号で置き換える。

#### 【0043】

次に、第2のフレーム目の画像から第17のフレーム目の画像まで計16枚の画像に対して、上記の第1のフレーム目と同じ操作を行い、それぞれの画像を4×4画素に分割したブロック領域を静止画用コードブックのベクトルのコード番



号で置き換える。

【0044】

静止画用コードブックのコード番号で表現された各フレームの画像の同じアドレスが示すブロックのコード番号について、1フレーム目のコード番号と2フレーム目のコード番号の差分をとり、次に1フレーム目と3フレーム目のコード番号の差をとる。このように、1フレーム目の画像のコード番号と17フレーム目の画像の同じアドレスのコード番号の差分をとり、アドレス順に並び変える。

【0045】

つまり、静止画ベクトル量子化を行った17フレーム分の画像の場合、1フレーム目の画像のコード番号と2フレーム目の画像のコード番号の差を新しく作られるデータ列の0番地の位置におき、1フレーム目の画像のコード番号と3フレーム目の画像のコード番号の差をデータ列の1番地の位置に置く。

【0046】

このように、1フレーム目のコード番号と16フレーム分のコード番号のそれぞれの差に対して並び代えが行われ、最後の1フレーム目の画像のコード番号と17フレーム目の画像のコード番号の差はデータ列の15番地に置かれることになる。ここで新しく生成された16個のコード番号列を含む集合をベクトルとする。

【0047】

この操作を静止画ベクトル量子化を行った画像のすべての画面に対して行くと、 $640 \times 480$ 画素の画像の場合でかつ一つのブロック領域を $4 \times 4$ 画素とした場合、 $160 \times 120 = 19200$ 個のブロックが生成されるので、19200個のベクトルが新しく生成される。

【0048】

次に、ここでは16個のデータで構成された2048個のベクトルからなる時間軸コードブックを用意しておき、先に生成された19200個のベクトルの内ある一つのベクトルの個々のデータと時間軸コードブックを構成するそれぞれのベクトルの個々のデータとの差分絶対値を演算し、各データにおける差分絶対値を加算し、その加算結果がもっとも小さい値を示したベクトルの時間軸コードブ

ック内のアドレスを特定する。

【0049】

同様の操作を19200個すべてのベクトルに対して行い、データ列で表現されていたベクトル列を時間軸コードブックの時間軸コード番号列で表現する。

【0050】

この段階で、17フレーム分の640×480画素分の画像は、ここでは11bit表現された19200個のデータからなるデータ列で表現されている。従って、16フレーム分の動画像を転送する場合、あらかじめ静止画用コードブック、時間軸コードブックを送っておき、その後時間軸コードブックのコード番号列で表現されたリストと1フレーム目の画像が静止画用コードブックのコード番号で表現されたデータ列を送ればよく、41779200bitの情報量を有する17フレーム分の画像を0.01倍にあたる422400bitの情報量を有する時間軸コードブックの時間軸コード番号列にまで画質を保持したまま圧縮することが出来る。

【0051】

このようにして転送されてきたデータをもとに、もとの16フレーム分の画像を再現することが出来る。まず、転送されてきた時間軸コード番号列の個々のコード番号に対応するコードをあらかじめ転送されている時間軸コードブックから参照し順番に配列する。ここでは、時間軸コードブックの1つのベクトルは16個のデータから構成されているために、時間軸コード番号列には19200個のデータがあるが、上記の処理により順番に配列されたデータ列は307200個のデータから構成される。

【0052】

次に、配列されたデータ列をここでは16個おきにデータを16分割し、19200個のデータからなるデータ列を生成し、1フレーム目のコード番号と各データ列を構成しているコード番号差分との加算を行い、データ列を再構成する。

【0053】

次に、1フレーム目のコード番号と16個に分割され再構成した静止画デコード用データ列の個々のコード番号に対応するコードをあらかじめ転送されている

静止画用コードブックから参照し、参照してきた画像データをはめこみもとの画像を再生する。

【0054】

これにより、もとの17フレーム分の画像を再生することが出来る。

【0055】

ここでは、1つのベクトルを構成するデータを16個だとし、静止画用コードブックが2048ベクトルで構成されているとし、時間軸コードブックが2048ベクトルで構成されているとしたが、これは実施例を述べるにあたり分かりやすくするためであり、ここで述べた数値に固定されるべきでないのは言うまでもない。また、ここでは17フレーム分の画像を時間軸ベクトル量子化により圧縮を行うことにしたが、圧縮を行うフレームの枚数を17フレームに限定する必要はなく、必要な枚数のフレームをベクトル量子化により圧縮を行ってよいことは言うまでもない。また、ここでは動画像を用いた実施例を示しているが、動画像に限らずに音声信号を対象にしても本発明の効果になんら変わりはない。

【0056】

(実施例4)

本実施例では、図10に示すように640×480画素からなり、1画素当たりの輝度値を8bit表現された動画像に対して圧縮伸長を行う方法について述べる。ここでは、17フレーム分の画像に対して圧縮を行っている。

【0057】

あらかじめ準備しておいた1枚の画像に対して、ベクトル量子化を行うための静止画用コードブックと呼ばれるデータベースを用意しておく。ここでは、この静止画用コードブックは16個のデータからなるデータ列を1つのベクトルとし、そのベクトルを複数組み合わせたものである。ここでは、2048ベクトル分を組み合わせ構成している。

【0058】

まず、第1フレーム目の画像を前記静止画用コードブックを用いて、ベクトル量子化を行う。ここでは、第1フレーム目の画像をX方向に4画素、Y方向に4画素ずつ区切ったブロック領域をそれぞれ入力ベクトルと定義し、その一つの入

カベクトルを構成している個々のデータと、静止画用コードブックを構成している各ベクトルの個々のデータとの差分絶対値をもとめ、そのそれぞれの結果を加算し、その加算結果を比較することにより、入力ベクトルにもっとも似通ったベクトルのコード番号を特定する。

【0059】

この操作を前記入力ベクトル全てに対して行い、第1フレーム目の画像を4×4画素に分割したブロック領域を静止画用コードブックのベクトルのコード番号で置き換える。

【0060】

次に、第2のフレーム目の画像から第17のフレーム目の画像まで計16枚の画像に対して、上記の第1のフレーム目と同じ操作を行い、それぞれの画像を4×4画素に分割したブロック領域を静止画用コードブックのベクトルのコード番号で置き換える。

【0061】

静止画用コードブックのコード番号で表現された各フレームの画像の同じアドレスが示すブロックのコード番号について、1フレーム目のコード番号と2フレーム目のコード番号の差分をとり、次に2フレーム目と3フレーム目のコード番号の差をとる。このように、16フレーム目の画像のコード番号と17フレーム目の画像の同じアドレスのコード番号の差分をとり、アドレス順に並び変える。

【0062】

つまり、静止画ベクトル量子化を行った17フレーム分の画像の場合、1フレーム目の画像のコード番号と2フレーム目の画像のコード番号の差を新しく作られるデータ列の0番地の位置におき、2フレーム目の画像のコード番号と3フレーム目の画像のコード番号の差をデータ列の1番地の位置に置く。

【0063】

このように、となりあうフレーム同士のそれぞれのコード番号の差に対して並び代えが行われ、最後の16フレーム目の画像のコード番号と17フレーム目の画像のコード番号の差はデータ列の15番地に置かれることになる。ここで新しく生成された16個のコード番号列を含む集合をベクトルとする。

## 【0064】

この操作を静止画ベクトル量子化を行った画像のすべての画面に対して行くと、 $640 \times 480$ 画素の画像の場合でかつ一つのブロック領域を $4 \times 4$ 画素とした場合、 $160 \times 120 = 19200$ 個のブロックが生成されるので、 $19200$ 個のベクトルが新しく生成される。

## 【0065】

次に、ここでは16個のデータで構成された2048個のベクトルからなる時間軸コードブックを用意しておき、先に生成された19200個のベクトルの内ある一つのベクトルの個々のデータと時間軸コードブックを構成するそれぞれのベクトルの個々のデータとの差分絶対値を演算し、各データにおける差分絶対値を加算し、その加算結果がもっとも小さい値を示したベクトルの時間軸コードブック内のアドレスを特定する。

## 【0066】

同様の操作を19200個すべてのベクトルに対して行い、データ列で表現されていたベクトル列を時間軸コードブックの時間軸コード番号列で表現する。

## 【0067】

この段階で、17フレーム分の $640 \times 480$ 画素分の画像は、ここでは11 bit 表現された19200個のデータからなるデータ列で表現されている。従って、16フレーム分の動画像を転送する場合、あらかじめ静止画用コードブック、時間軸コードブックを送っておき、その後時間軸コードブックのコード番号列で表現されたリストと1フレーム目の画像が静止画用コードブックのコード番号で表現されたデータ列を送ればよく、 $41779200$  bit の情報量を有する17フレーム分の画像を0.01倍にあたる $422400$  bit の情報量を有する時間軸コードブックの時間軸コード番号列にまで画質を保持したまま圧縮することが出来る。

## 【0068】

このようにして転送されてきたデータをもとに、もとの16フレーム分の画像を再現することが出来る。まず、転送されてきた時間軸コード番号列の個々のコード番号に対応するコードをあらかじめ転送されている時間軸コードブックから

参照し順番に配列する。ここでは、時間軸コードブックの1つのベクトルは16個のデータから構成されているために、時間軸コード番号列には19200個のデータがあるが、上記の処理により順番に配列されたデータ列は307200個のデータから構成される。

#### 【0069】

次に、配列されたデータ列をここでは16個おきにデータを16分割し、19200個のデータからなるデータ列を生成し、1フレーム目のコード番号と1つめのデータ列のデータとの加算を行い、2フレーム目の画像の静止画デコード用データ列を生成する。次に、2フレーム目の画像の静止画デコード用データ列のデータと2つめのデータ列のデータとの加算を行い、3フレーム目の画像の静止画デコード用データ列を生成する。同じ操作を残りのデータ列についても順番に行い、1フレーム目を除く16フレーム分の静止画デコード用データ列を再構成する。

#### 【0070】

次に、1フレーム目のコード番号と16個に分割され再構成した静止画デコード用データ列の個々のコード番号に対応するコードをあらかじめ転送されている静止画用コードブックから参照し、参照してきた画像データをはめこみもとの画像を再生する。

#### 【0071】

これにより、もとの17フレーム分の画像を再生することが出来る。

#### 【0072】

ここでは、1つのベクトルを構成するデータを16個だとし、静止画用コードブックが2048ベクトルで構成されているとし、時間軸コードブックが2048ベクトルで構成されているとしたが、これは実施例を述べるにあたり分かりやすくするためであり、ここで述べた数値に固定されるべきでないのは言うまでもない。また、ここでは17フレーム分の画像を時間軸ベクトル量子化により圧縮を行うことにしたが、圧縮を行うフレームの枚数を17フレームに限定する必要はなく、必要な枚数のフレームをベクトル量子化により圧縮を行ってよいことは言うまでもない。また、ここではフレーム間のコード番号の差分を取る方法を隣

り合うフレーム同士のコード番号差分をとっているが、必ずしも隣り合うフレーム間のコード番号の差分を取る必要はないことは言うまでもない。また、ここでは動画像を用いた実施例を示しているが、動画像に限らずに音声信号を対象にしても本発明の効果になんら変わりはない。

#### 【0073】

##### (実施例5)

本実施例は、時間軸方向に画像データを圧縮する手法については実施例1～3で述べたいずれの手法を用いてもよく、圧縮した画像を再生する際により高画質な再生画像を得るために発明した手法の具体例について述べる。ここでは、実施例1に新たな手法を加えた例を示すため、実施例1と同じ処理を行っている部分については省略する。

#### 【0074】

入力された16枚の画像に対して第1の実施例で述べた処理を行い、16フレーム分の画像をひとつの時間軸コードブックの時間軸コード番号列で表現する。この時間軸コード番号列をもとに、16フレーム分の画像を再現するのだが、時間軸コード番号列の個々のコード番号に対応するコードをあらかじめ転送されている時間軸コード番号から参照し、順番に配列する。

#### 【0075】

次に、配列されたデータ列を16個おきにデータを16分割し、16個の静止画デコード用データ列を生成する。第1の実施例では、この分割された静止画デコード用データ列の個々のコード番号に対応するコードをあらかじめ転送されてきている静止画用コードブックから参照し、参照してきた画像データをはめ込むことでもとの画像を再生するのだが、この手法だと1枚目のフレームの再生画像は画質の高いものが得られるが、フレームがすすむにつれ時間軸方向のベクトル量子化による量子化誤差が目立つようになり、画質が劣化してくる。その画質の劣化を防ぎかつ圧縮率を同等にするために、

静止画デコード用データ列の個々のコード番号に対応するコードをあらかじめ転送されてきている静止画用コードブックから参照し、参照してきた画像データをはめ込む時に、ベクトル量子化を行う時に定義したブロックの所望のアドレス

ごとに、時間軸方向にずらして画像データをはめ込むことで、時間軸方向のベクトル量子化により生じる量子化誤差を時間軸方向に分散させることができるのである。

【0076】

これにより、フレームがすすむごとに目立っていた量子化誤差が目立たなくなり、全体的に画質が向上するという利点が得られる。

【0077】

(実施例6)

コードブック方式動画像圧縮装置の構成図を図11に示す。

【0078】

画像入力装置601は画像を連続して取り込む装置である。ここでは例えば1 pixelが8 bitの階調を持った640×480 pixelの画像を1秒間に30枚取り込む。

【0079】

コードブック記憶装置602には4×4 pixelの画像が多数登録されていて、それぞれのブロックにはアドレスが割り当てられている。

【0080】

コードブック方式による圧縮装置603は画像入力装置601で得られた画像一枚一枚に対して次の操作を行う。

【0081】

画像の左上を起点としてそこから右方向へ向かって順次4×4 pixelのブロックを取り出す。右端まで取り出したならば取り出す位置を1ブロック分下にずらして左端から取り出して行き、それを繰り返す事により全画面分のブロックを取り出す。取り出されたブロックに対して、コードブック記憶装置602に登録されているブロックから最もパターンの似ているものを選び出し、そのアドレスをコードとして出力する。640×480 pixelの画像を処理する場合、19200個のブロックが取り出されて処理されるので、出力コードも19200個出力される。これをこれからコード列と呼ぶことにする。

【0082】



コード列記憶装置 604 は、ある時刻  $t$  の画像をコードブック方式による圧縮装置 603 にて処理して出力されるコード列を記憶しておく装置である。コード列比較装置 605 は、コード列記憶装置 604 に記憶されているある時刻  $t$  の画像のコード列と次の時刻  $t+1$  の画像のコード列に対して各コードの差分絶対値を取り、差分絶対値が転送閾値よりも大きい場合、コードの番号（コードの順番の番号）とそのコードの値を出力する。

【0083】

コード列比較装置 605 から出力されるデータは転送媒体 606（通信チャンネルまたは記録媒体）に与えられる。

【0084】

（実施例 7）

次にコードブック方式動画像圧縮処理をソフトウェアで実現する実施例 7 に付いて説明する。

【0085】

図 12 は CPU を含むコードブック方式動画像圧縮プロセッサの構成を示すものである。画像入力装置 601 および転送媒体 606 は図 11 に示すものと同じである。画像入力装置 601 は画像取り込み用カメラのみならず、半導体メモリ装置、ハードディスクやフロッピー・ディスクによって実現することも出来る。コードブック方式による圧縮装置 603 も図 11 に示すものと同じである。

【0086】

CPU 7-101 に付随するメモリ 7-102 には、コードブック記憶エリア 7-103、コード記憶エリア 7-104 が設けられている。

【0087】

コードブック記憶エリア 7-103 はコードブック方式による圧縮装置 603 で使用するコードブックを記憶しておくものである。

【0088】

コード列記憶エリア 7-104 はコードブック方式による圧縮装置 603 から出力される出力コードを記憶しておくもので、時刻  $t$  の画像をコードブック方式による圧縮装置 603 で圧縮したコードが記憶される。

【0089】

ポインタ・カウンタ・エリア7-105はコード列記憶エリアに対する読み出し・書き込みを行う位置を示すポインタを記憶する。

【0090】

転送閾値記憶エリア7-106は転送するデータを決定するための閾値が記憶されている。

【0091】

図13はCPU7-101が実行する圧縮処理の手順を示している。

【0092】

まず、コードブック記憶エリア7-103からコードブック方式圧縮装置603へコードブックのデータを転送する（ステップ7-201）。

【0093】

転送閾値記憶エリア7-106に転送閾値を設定する（ステップ7-202）。

【0094】

ポインタ・カウンタ・エリア7-105のポインタがイニシャライズされる（ステップ7-203）。

【0095】

画像入力装置601からマクロブロックを読み込む。例えば4×4 pixelのマクロブロックを読み込む（ステップ7-205）。

【0096】

読み込まれたマクロブロックをコードブック方式圧縮装置603に送り、圧縮コードを得る（ステップ7-206）。

【0097】

コードブック方式圧縮装置603のから出力される出力コードとコード列記憶エリア7-104に記憶されているコードの差分絶対値を計算する（ステップ7-207）。計算された差分絶対値が転送閾値記憶エリア7-106に記憶されている転送閾値よりも大きい場合（ステップ7-208でYES）には出力コードとポインタの値を転送媒体606に出力する。

## 【0098】

計算された差分絶対値が転送閾値記憶エリア7-106に記憶されている転送閾値よりも小さい場合（ステップ7-208でNO）にはステップ7-209を実行せずに、コードブック方式圧縮装置603のから出力される出力コードをコード列記憶エリア7-104に記憶する。

## 【0099】

ポインタ・カウンタ・エリア7-105のポインタをインクリメント（ステップ7-211）し、ステップ7-204を経て7-205に戻る。

## 【0100】

上述した処理は画像入力装置601から得られた画像中のすべてのマクロブロックを処理するまで続けられる。画像中のすべてのマクロブロックを処理する（ステップ7-204でYES）となると、動画像1フレーム分に対する圧縮処理は全て終了する。

## 【0101】

## （実施例8）

図14は初期のコードブックとして輝度値が連続的に変化したもの（とり得る値の最小値から最大値まで連続的に変化するもの）と、輝度値128で一定のものと、ランダムパターンのときに、このコードブックを用いて画像を圧縮・伸長したときのPSNR（peak signal to noise ratio）特性の違いである。

## 【0102】

## （実施例9）

図15は、マップ上（仮想の2次元平面）での、書き換えを行う範囲を1次元的に適用させ更新回数の増加とともに範囲を減少させていく方法と2次元的に適用させ更新回数の増加とともに範囲を減少させていく方法のPSNR特性の違いであり、範囲の初期値を1, 2, 3, 4, 5とし、更新回数の増加と共に減少させていったものである。本実施例では、範囲の初期値の最大値を5としたが、5である必要がないことは言うまでもない。

## 【0103】

## （実施例10）

図16は、利得を更新回数の増加にかかわらず初期値から減少させないときと更新回数の増加とともに初期値から減少させるときのPSNR特性の違いである。

#### 【0104】

図17は、更新回数160000回のときの利得とPSNR特性の関係を表したものである。

#### 【0105】

##### (実施例11)

図18は、最適化で得られた3種類のコードブックAとBとCの特徴を一つのコードブックA+B+Cにまとめる手法を示している。A+B+Cのコードブックサイズは同様に1024であるが、その中に含まれるパターンはA、BおよびCに含まれるパターンを間引いて合成したものである。ここでは、Kohonenの自己組織化マップで得られたコードブックは似たようなパターンがマップ上に近接して存在するため、この特性を生かしA、BおよびCのマップ上の左上から順番に一つおきにそのままパターンを取りだし、これをA+B+Cとして再構成しているだけである。これによりそれぞれの特徴を残したコードブックが新たに生成された。

#### 【0106】

図19にA、B、C、A+B+Cそれぞれのコードブックを用いて画像A、B、Cに対してベクトル量子化した後復元した画像のPSNRを示す。もともとのA、BおよびCのPSNR特性に対しA+B+CのPSNR特性は全ての画像に対して匹敵しており、かつ両者の特性を兼ね備えたものになっている。このように特徴的な画像を組み合わせることでコードブックを作成することで様々な画像に対して耐性を有するコードブックを造り上げることは可能であろうと考えられる。

#### 【0107】

上記実施例では同じコードブックサイズの3種類のコードブックを1つにまとめる手法を示したが、3種類のみというわけではなく、複数のコードブックをまとめて1つにすることもでき、コードブックサイズの異なる複数のコードブックでもまとめて1つにすることができる。

【0108】

(実施例 12)

図 20 は、二つのベクトルが類似しているか否かを数値化した類似度を求める関数としてマンハッタン距離を、特徴量としてベクトルデータの要素の総和を利用した場合のベクトル量子化装置の構成と、演算が省略される様子とを説明するものである。

【0109】

なお、類似度としてマンハッタン距離を用いた場合、距離が小さいほうが類似度は大きい。

【0110】

最初に、演算が省略できる原理を説明する。

【0111】

$n$  をベクトルデータの次元数、 $k$  をテンプレート記憶装置内のテンプレートベクトルデータの総数とする。

【0112】

入力ベクトルデータ:  $I = (I_1, I_2, \dots, I_n)$

記憶装置内のベクトルデータ:  $T_m = (T_{m1}, T_{m2}, \dots, T_{mn})$   
 $(m = 1, 2, \dots, k)$

について、 $I$  と  $T_m$  の類似度をマンハッタン距離  $M_m$  を

$$M_m = \sum_{l=1}^n |I_l - T_{ml}|$$

で定義すると、入力ベクトルデータ  $I$  に最も類似している記憶装置内のベクトルデータ  $T_m$  を検索する操作とは、全ての  $M_m$  ( $m = 1, 2, \dots, k$ ) の中で  $M_m$  が最も小さくなる  $m$  を探し出す操作に相当する。

【0113】

ここで、 $I$  ベクトルデータ要素の総和と  $T_m$  のベクトルデータ要素の総和の差の絶対値  $D_m$

$$D_m = \sum_{l=1}^n I_l - \sum_{l=1}^n T_{ml}$$

$$D_m = \left| \sum_{l=1} I_l - \sum_{l=1} T_{ml} \right|$$

と、マンハッタン距離  $M_m$  との間には

$$D_m \leq M_m \quad (1)$$

という関係が成り立つことに注目すると、以下のような議論が成り立つ。

【0114】

テンプレート記憶装置内のあるテンプレートベクトルデータ  $T_m$  と入力ベクトルデータ  $I$  のマンハッタン距離  $M_m$  が判明しているとする。テンプレート記憶装置内にあって  $T_m$  とは別な任意のテンプレートベクトルデータ  $T_o$  と入力ベクトルデータ  $I$  について、各々のベクトルデータの要素の総和を取り、該総和どうしの差の絶対値  $D_o$  を計算し、 $D_o > M_m$  となっている場合を仮定する。

【0115】

上述の  $D_o$  と  $M_o$  の関係(1)により、常に  $M_o > M_m$  が成り立つ。よって、 $T_o$  と  $I$  のマンハッタン距離  $M_o$  を計算せずとも、 $D_o$  を計算するだけで、 $T_o$  は  $T_m$  に比べて  $I$  に類似していないことがわかる。つまり、 $T_o$  を検索対象から除外できることになる。

【0116】

以上だけ見ると、 $M_o$  を計算しなくても、 $D_o$  を計算する必要があるのならば、検索に関わる処理が減少していないように見える。しかし、 $D_o$  の演算においてテンプレートベクトルデータの特徴量である要素の総和

$$\sum_{l=1}^n T_{ml}$$

は既に計算されて特徴量記憶装置に記憶されているから、 $D_o$  の算出は入力ベクトルデータの特徴量と、特徴量記憶装置内に記憶されている特徴量との差をとるだけで済む。また、入力ベクトルデータの特徴量の算出は、一入力ベクトルデータに対し一回で済む。このため、 $D_o$  の算出は  $M_o$  に比べて非常に少ない演算で行うことができる。

【0117】

なお、特徴量として要素の総和を用いるのと、要素の平均値を用いるのは等価である。なぜならば、要素の平均値とは要素の総和を要素数で割っただけのものに過ぎないからである。

【0118】

次に、図20を使って具体的に説明する。

入力ベクトルデータ1210、テンプレートベクトルデータ1211ないし1218は、例として6次元のベクトルデータとなっているが、任意の次元でかまわない。特徴量記憶装置1204にはテンプレート記憶装置1201内のテンプレートベクトルデータ1211ないし1214の特徴量が記憶されている。図20中ではテンプレートベクトルデータ1211ないし1214と、すぐ左側に位置する特徴量1215ないし1218は対応している。つまり、テンプレートベクトルデータ1211の特徴量は1215、テンプレートベクトルデータ1212の特徴量は1216となる。

【0119】

また、マンハッタン距離演算装置1202は12bで示されるテンプレート記憶装置から転送されたテンプレートベクトルデータと、入力ベクトルデータ1210とのマンハッタン距離を求め、マンハッタン距離12cを出力する。最小マンハッタン距離記憶装置1203は、今までに見つかった最小のマンハッタン距離を記憶する。

【0120】

特徴量演算装置1205は入力ベクトルデータ1210のベクトルデータ要素の総和を演算し、保存し、出力する。特徴量差分演算装置1206は、特徴量演算装置で求められた特徴量12fと特徴量記憶装置1204から読み出された特徴量12dとの差分を演算し、出力する。演算省略判定装置1207は最小マンハッタン距離記憶装置1203で記憶されている、今までにわかっている最小のマンハッタン距離12hと特徴量差分演算装置1206で演算された特徴量の差分12gとを利用して、テンプレートベクトルデータと入力ベクトルデータのマンハッタン距離を演算する必要があるか否かを判別する。

【0121】

ベクトル量子化の手順を次に示す。演算の省略の様子も同時に説明する。

1) 入力ベクトルデータの特徴量を特徴量演算装置 1205 によって演算し、保存する。この特徴量は、一つの入力に対し、一度だけ演算すればよい。

2) 入力ベクトルデータと第一番目のテンプレートベクトルデータ 1211 とのマンハッタン距離  $M_1$  をマンハッタン距離演算装置 1202 によって演算する。マンハッタン距離はまだ一つしか演算していないので、現在までで見つかった最小のマンハッタン距離  $\min M$  は  $M_1$  となる。したがって  $M_1$  を最小マンハッタン距離記憶装置 1203 に記憶する。

【0122】

$$M_1 = \sum_{l=1}^6 |I_1 - T_{1l}| = 70$$

である。また、 $\min M = M_1$  である。

3) 第二番目のテンプレートベクトルデータ 1212 の特徴量 1216 と入力ベクトルデータ 1210 の特徴量 12f の差の絶対値  $D_2$  を特徴量差分演算装置 1206 で求める。

【0123】

$$D_2 = \left| \sum_{l=1}^6 I_1 - \sum_{l=1}^6 T_{2l} \right| = 4$$

である。

4) 手順 3) で求めた特徴量の差の絶対値  $D_2$  と現在までに見つかった最小のマンハッタン距離  $\min M$  をもとに、第二番目のテンプレートベクトルデータ 1212 と、入力ベクトルデータ 1210 とのマンハッタン距離を演算する必要があるか、演算省略判定装置 1207 で判断する。 $D_2 < \min M$  であるから、第二番目のテンプレートベクトルデータ 1212 と、入力ベクトルデータ 1210 とのマンハッタン距離演算は省略できない。

5) 手順 4) により、第二番目のテンプレートベクトルデータ 1212 と、入力ベクトルデータ 1210 とのマンハッタン距離演算が必要と判定されたので、こ



の演算を行う。第二番目のテンプレートベクトルデータ 1212 と、入力ベクトルデータ 1210 とのマンハッタン距離  $M_2$  をマンハッタン距離演算装置 1202 で演算する。 $M_2 = 22$  である。 $M_2$  は、今まで見つかった最小のマンハッタン距離  $\min M$  よりも小さいので、 $\min M$  の値を  $M_2$  に更新する。 $\min M = M_2$  になる。つまり最小マンハッタン距離記憶装置 1203 には  $M_2$  が記憶される。

6) 第三番目のテンプレートベクトルデータ 1213 の特徴量 1217 と、入力ベクトルデータ 1210 の特徴量 12f との差の絶対値  $D_3$  を特徴量差分演算装置 1206 で演算する。 $D_3 = 46$  である。

7) 手順 6) で求めた特徴量の差の絶対値  $D_3$  と現在までに見つかった最小のマンハッタン距離  $\min M$  をもとに、第三番目のテンプレートベクトルデータ 1213 と、入力ベクトルデータ 1210 とのマンハッタン距離を演算する必要があるか、演算省略判定装置 1207 で判断する。 $D_3 > \min M$  であるから、第三番目のテンプレートベクトルデータ 1213 と、入力ベクトルデータ 1210 とのマンハッタン距離演算は省略できる。

8) 手順 7) で第三番目のテンプレートベクトルデータ 1213 と、入力ベクトルデータ 1210 とのマンハッタン距離演算は省略できると判明したので、このマンハッタン距離演算を省略して、第四番目のテンプレートベクトルデータ 1214 の特徴量 1218 と入力ベクトルデータ 1210 の特徴量 12f との差分の絶対値  $D_4$  を特徴量差分演算装置 1206 で演算する。 $D_4 = 157$  である。

9) 手順 8) で求めた特徴量の差の絶対値  $D_4$  と現在までに見つかった最小のマンハッタン距離  $\min M$  をもとに、第四番目のテンプレートベクトルデータ 1214 と、入力ベクトルデータ 1210 とのマンハッタン距離を演算する必要があるか、演算省略判定装置 1207 で判断する。 $D_4 > \min M$  であるから、第四番目のテンプレートベクトルデータ 1214 と、入力ベクトルデータ 1210 とのマンハッタン距離演算は省略できる。

10) 以下、同様の操作を繰り返す。

【0124】

以上の手順の中にも示された通り、図 20 に記載の構成のベクトル量子化装置

で、入力ベクトルデータにもっとも類似したテンプレートベクトルデータを検索する操作を行う際、検索に必要な類似度の演算を減少させることができた。

#### 【0125】

また、図20では、テンプレート記憶装置内のテンプレートベクトルデータ1211から1214は任意の順で並んでいたが、特徴量の順で並べておいても良い。テンプレートベクトルデータが特徴量順に並んでいる場合、テンプレートベクトルデータを検索する際、入力ベクトルデータ1210と特徴量の最も近いテンプレートベクトルデータから検索することが容易になる。二つの特徴量の近いベクトルデータは、類似度も大きい傾向があるので、入力ベクトルデータと特徴量が最も近いテンプレートベクトルデータから検索を行うことで、類似度演算はより多く省略される。

#### 【0126】

本例では、ベクトル量子化装置が取り扱うベクトルデータの次元を6次元としたが、ベクトルデータの次元は任意で良い。

#### 【0127】

なお、図20のベクトル量子化装置は、専用のハードウェアで実現することも可能であるし、コンピュータ上のプログラムによってソフトウェアで実現することもできる。

#### 【0128】

図20において、ベクトルデータの特徴量としてベクトルデータの分散を使いたい場合、特徴量演算装置1205と、特徴量省略判定装置1207を変更する。また、類似度を求める関数を変更したい場合、マンハッタン距離演算装置1202と、最小マンハッタン距離記憶装置1203と、演算省略判定装置1207とを変更する。

#### 【0129】

##### (実施例13)

実施例13では、第12実施例と同様なベクトル量子化装置の入力を画像データとし、本発明のベクトル量子化装置を画像圧縮装置に応用した例を、図21ないし図22を使って説明する。また、ベクトル量子化装置の入力を画像データに

した場合の、画像データ特有な特徴量について説明し、幾つかなの特徴量を組み合わせてベクトル量子化処理の演算量を減少させることが可能であることを示す。

#### 【0130】

まず、ベクトル量子化装置を使って画像圧縮を行う例を図21を使って説明する。

#### 【0131】

入力画像13-101は、画素と呼ばれる要素が集まって構成されている。画素は輝度値、あるいは色差信号などの情報を持っている。入力画像13-101から複数画素で構成されるブロックを取り出したのが、入力画像ブロック13-103である。図21では入力画像ブロックの大きさとして4×4画素を選んでいるが、入力画像ブロックの大きさは何であっても構わない。

#### 【0132】

入力画像ブロック13-103は、前述の通り複数の画素を持っているから、各々の画素が持つ輝度値を集めて、ベクトルデータとすることができる。これをベクトル量子化装置の入力ベクトルデータとする。

#### 【0133】

人間の視覚特性上、入力画像13-101上の幾つかなの画像ブロックは殆ど見た目では同じに見える場合がある。こういった、同じに見える複数の画像ブロックを、より少ない数の画像ブロックで代表させることが可能である。画像ブロックコードブック13-104は、入力画像13-101上の多数の画像ブロックを代表する画像ブロックである。テンプレートベクトルデータは、画像ブロックコードブック13-104内の画像ブロック各々の画素が持つ輝度をベクトルデータとしたものである。

#### 【0134】

入力画像13-101全体を画像ブロックとして分割し、各々の画像ブロックをベクトル量子化装置の入力ベクトルデータとし、入力ベクトルデータに類似するテンプレートベクトルデータを検索して、テンプレートベクトルデータの番号13-105のみを転送することで、画像の圧縮が可能である。

#### 【0135】

なお、以上の例では画素の輝度値をベクトルデータとしたが、輝度値だけでなく、画素が持つ数値化された情報は何でもベクトルデータにできる。また、画像を画像ブロックとして取り出した後、離散余弦変換をはじめとする様々な変換を掛け、変換後の画像からベクトルデータを作って、ベクトル量子化装置の入力としても良い。

## 【0136】

入力画像 13-101 から入力ベクトルデータを作成してしまえば、類似度を求める関数としてマンハッタン距離を、特徴量としてベクトルデータの要素の総和を使った場合、ベクトル量子化の手順は実施例 12 と全く同一である。よって、ここでは詳細な手順の説明を省く。なお、入力ベクトルデータが画像データの場合、特徴量としてベクトルデータの要素の総和を用いても、画像データを周波数分解した際の直流成分を用いても、二つは等価である。何故ならば、前記直流成分は、ベクトルデータの要素の総和にある係数を掛けたものに過ぎないからである。

## 【0137】

この方法によって、画像データに対しベクトル量子化の操作を行った時に、マンハッタン距離演算がどれほど省略できるのかを示したのが図 22 である。図 22 は、屋外風景写真、屋内風景写真、人物写真を入力画像とした時に、様々なテンプレートベクトルデータ群に対して、テンプレートベクトルデータ群の中でマンハッタン距離を演算しなければならないテンプレートベクトルデータの割合を表している。図 22 によると、テンプレートベクトルデータの中で、入力ベクトルデータとマンハッタン距離を演算しなければならないのはせいぜい 14% 以内となっていることが分かり、本発明の有用性が示されている。

入力ベクトルデータが画像データである場合、ベクトルデータの特徴量として、画像特有の性質を使うことが可能である。以下に説明する。

## 【0138】

画像データに対して、類似度を演算する関数をマンハッタン距離、特徴量としてベクトル要素の総和とした場合の問題点を図 23 を使って示す。入力ベクトルデータの元となる画像ブロック 13-301 と、テンプレートベクトルデータに

対応する画像ブロック 13-303 は、全く上下左右反転したブロックとなっており、見た目では全く異なるものである。実際に、画像ブロック 13-301 と、画像ブロック 13-303 との画素の輝度値を取り出して、それぞれベクトルデータにし、マンハッタン距離演算をすると、画像ブロック 13-303 は、画像ブロック 13-302 に比較して画像ブロック 13-301 と類似していないことが明らかになる。しかし、特徴量である、ベクトルデータの要素の総和は、画像ブロック 13-302 と 13-303 とで等しい。このことは、特徴量としてベクトル要素の総和を使っただけでは、無駄な類似度演算が多く残ることを意味する。

#### 【0139】

この問題を解決するためには、特徴量として、ベクトルデータの要素の一部を画素の明暗が反転するように操作した後のベクトルデータの総和を使うという手段が考えられる。図 24 はベクトルデータの一部を反転させた場合、画像ブロックがどのように変化するかを表している。初期画像ブロック 13-401 に対し、反転パターン 13-402 から 13-404 の黒で塗りつぶされた部分の画素が明暗反転するように操作した結果が、反転後画像ブロック 13-405 から 13-407 である。ベクトルデータ要素の総和が等しい場合でも、ベクトルデータ要素の一部を画素の明暗が反転するように操作してから、ベクトルデータの要素の総和を求めて特徴量とすると、特徴量の差が現れてくる例が図 25 である。画像ブロック 13-501 と 13-502 は、画素の輝度値の総和は全く同じである。よって、ベクトルデータの要素の総和、即ちブロック内の画素の輝度値の総和を特徴量としても二つは全く区別できない。反転パターン 13-503 の黒塗りの部分を明暗反転することで、反転後の画像ブロックについての画素の輝度値の総和 13-506 と 13-507 は全く異なったものとなる。このことを利用して無駄な演算を減少させることができる。

#### 【0140】

以上の例からもわかるように、ある特徴量を使ったときには省略できない類似度の演算が、別の特徴量を使うと省略できる場合がある。この場合、特徴量を二つ以上利用して、演算量を更に減少させることができる。

## 【0141】

具体的には、ある特徴量でテンプレートベクトルデータの中から類似度を演算しなければならない範囲を絞り込み、別の特徴量で更に範囲を絞り込むという、二段階で特徴量を使う方法と、同時に二つの特徴量を使って、特徴量の差が大きいほうを演算省略の判断に使う方法などが挙げられる。

## 【0142】

ある画像から、画像ブロックを取り出してきて、ベクトル量子化装置の入力とする場合、上述の「ベクトルデータの要素の一部を画素の明暗が反転するように操作した後のベクトルデータの総和」の他にも、画像特有な特徴量がいくつか考えられる。ここでは、画像ブロックの四隅の画素を特徴量として使う方法と、画像ブロック上の画素の変化を特徴量として使う方法を説明する。

## 【0143】

最初に、画像ブロックの四隅の画素を特徴量として使う方法を説明する。画像ブロックを画像から取り出してきた場合に、滑らかに変化する画像に対しては、画像ブロックの四隅から特徴量ができる。図26では、画像ブロックの四隅を特徴量として使う方法を説明している。画像ブロック13-601は、ブロックの右下から左上方向に、輝度が明るくなる変化をしている。画像ブロック13-601の四隅13-603を取り出し、画素の輝度値を比較すると、画像ブロックがどのような方向に変化しているのかを把握することができる。画像ブロック13-602と、その四隅13-604は、画素の変化する方向が異なる場合の一例である。このように、画像ブロックの四隅の画素の間で大小比較し、大小を判別しすることで、画像ブロックの特徴を捉えることができる。四隅の画素の間での大小のパターンは幾通りもあるが、このパターンは番号付けして、値とすることができる。四隅の画素の間での大小のパターン13-605の番号13-606を特徴量として、本発明のベクトル量子化装置に組み込むことができる。

## 【0144】

図26では画像ブロックの大きさとして4画素×4画素としているが、この大きさは任意で良い。

## 【0145】

次に、画像ブロック上での画素の変化を特徴量として使う方法を説明する。画像から画像ブロックを幾つか切り出して来ると、画像ブロックの画素の変化の様子が似ているものがある。図27は、画素の変化の周期が同じである二つの画像ブロックを示している。画像ブロック13-701と13-702はいずれも左右方向にのみ画素の値が変化している。画像ブロック13-701の左右方向の画素の値の変化は、画像ブロック13-702の左右方向の画素の値の変化に係数を掛けて、変化の振幅を押さえたものに他ならない。このような状態を、画像ブロック13-701と画像ブロック13-702の、画像ブロック上での画素の変化のモードが同じであると言うことにする。画像ブロック上の画素の変化のモードは多数あるが、変化のモードに番号付けして値とすることができる。画素の変化のモードの番号を特徴量として、本発明のベクトル量子化装置に組み込むことができる。

## 【0146】

図27では、画像ブロックの大きさとして4画素×4画素としているが、この大きさは任意で良い。

## 【0147】

実施例13の最後は、第一の特徴量として画像ブロック上での画素の変化、つまり画像ブロック上の変化のモードを番号付けしたものと、第二の特徴量として画素の値の総和、つまりベクトルデータの要素の総和とを使った、本発明のデータ圧縮システムにおけるベクトル量子化装置の構成を、図28を使って説明する。なお、類似度は、マンハッタン距離によって求めている。図28中では、ベクトルデータを、相当する画像ブロックとして視覚的に表現している。

## 【0148】

入力ベクトルデータ13-801は、第一の特徴量である画像ブロック上の画素の変化モードによって分類される。変化モード判定装置13-802は、画素の変化モードの典型的なベクトルデータを集めた特徴テンプレート記憶装置13-803に記憶されたベクトルデータの情報を元に、入力ベクトルデータの第一の特徴量である画像ブロック上の画素の変化モードが何番であるか出力する13-804。テンプレートベクトルデータを記憶するテンプレート記憶装置と、テ

ンプレートベクトルデータの第二の特徴量であるテンプレートベクトルデータの要素の総和を記憶する特徴量記憶装置の内部では、第一の特徴量によってデータが並べられている。図 28 の例では、入力ベクトルデータの第一の特徴量は” 2 ”と判断されたので、テンプレート記憶装置と、特徴量記憶装置内との中から第一の特徴量が” 2 ”となる部分 13-820 のみが選択され、残りの部分は検索対象外となる。

## 【0149】

テンプレート記憶装置、特徴量記憶装置の中で、第一の特徴量が” 2 ”となる部分 13-820 に関して、実施 12 と同様の手順で検索を行う。入力ベクトルデータ 13-801 の第二の特徴量を特徴量演算装置 13-808 で演算し、その結果と、特徴量記憶装置内の特徴量 13-813 とを使って特徴量差分演算装置 13-809 ならびに演算省略判定装置 13-810 で、テンプレート記憶装置内の第一の特徴量が” 2 ”となるテンプレートベクトルデータの中で特にマンハッタン距離演算が必要の無いものを判断する。

## 【0150】

以上の例では、実施例 12 に挙げたような特徴量を単独で使う場合に比べて、より多くの演算が省略できている。第一の特徴量によってあらかじめ検索対象となるテンプレートベクトルデータを限定し、第二の特徴量で更に検索対象を絞り込むからである。

## 【0151】

これによって、特徴量の一つだけ使う場合に比べて、より演算の省略が可能になった。

## 【0152】

また、テンプレートベクトルデータは、第一の特徴量だけでなく、第二の特徴量によっても並べ、二次元的な配列にしておくことで、入力ベクトルデータ 13-801 に、第二の特徴量が似たテンプレートベクトルデータから検索を始められるので、より効率的に、入力ベクトルデータ 13-801 と類似のテンプレートベクトルデータを検索できる。

## 【0153】



なお、図 28 では、画像ブロックの大きさとして 4 画素×4 画素、ベクトルデータに直せば 16 次元のものを例に説明したが、この大きさは任意である。

【0154】

(実施例 14)

図 29 はコードブック・サーバシステムとその運用構成を示している。

【0155】

コードブック方式圧縮システム 1401 は、外部入力されたあるいは蓄積されているデータをコードブック方式により圧縮を行う装置である。

【0156】

コードブック・サーバ 1402 はコードブック蓄積装置 1411・コードブック管理装置 1412・コードブック登録装置 1413・コードブック生成装置 1414 を持ち、圧縮・伸張システムからのリクエストに応じてコードブックを転送する機能を持つ。

【0157】

コードブック方式圧縮システム 1401 においてデータ圧縮処理過程で使用するコードブックは、コードブック・サーバ 1402 より転送されたものを用いる。どのようなコードブックが必要であるかは、コードブック・サーバ 1402 に対するコードブック・リクエスト 1403 を転送媒体 1404 を介して送信する。

【0158】

コードブック・サーバ 1402 はコードブック・リクエスト 1403 に対する返答として、コードブック管理装置 1412 がリクエストされた物に合ったコードブックのデータをコードブック蓄積装置 1411 に蓄積されたコードブックの中から探し出し、転送媒体 1404 を介してコードブック方式圧縮システム 1401 にコードブックデータ 1405 を送信する。リクエストされた物に合致したコードブックがない場合には最も似通ったコードブックからリクエストに合ったコードブックをコードブック生成装置 1413 が生成する。この時、コードブック蓄積装置 1411 に蓄積されたコードブックを基に生成しても良いし、全く新たに生成しても良い。

## 【0159】

コードブック方式圧縮システム1401で生成される圧縮データは転送媒体1406を介してコードブック方式伸張システムに送信される。

## 【0160】

コードブック方式伸張システム1407においてデータ伸張処理過程で使用するコードブックは、コードブック・サーバ1402より転送されたものを用いる。どのようなコードブックが必要であるかは、コードブック・サーバ1402に対するコードブック・リクエスト1408を転送媒体1409を介して送信する。

## 【0161】

リクエストする内容は、コードブック方式圧縮システム1401で生成されたものを圧縮データと共に送信したリクエスト内容を使用しても良いし、コードブック方式伸張システム1407が独自に生成したリクエスト内容であっても良く、圧縮と伸張で異なるコードブックを使用しても差し支えない。

## 【0162】

コードブック・サーバ1402はコードブック・リクエスト1408に対する返答として、コードブック管理装置1412がリクエストされた物に合ったコードブックのデータをコードブック蓄積装置1411に蓄積されたコードブックの中から探し出し、転送媒体1409を介してコードブック方式伸張システム1407にコードブックデータ1410送信する。リクエストされた物と合致するコードブックがない場合には最も似通ったコードブックからリクエストに合ったコードブックをコードブック生成装置1413が生成する。この時、コードブック蓄積装置1411に蓄積されたコードブックを基に生成しても良いし、全く新たに生成しても良い。

## 【0163】

## 【発明の効果】

以上説明したように、請求項1に関わる発明によれば、ベクトル量子化により圧縮伸長したデータ列を表示装置に表示させる時に、位置をずらすことにより圧縮率を保ったまま質の高い再生画像を得ることが出来る。

【0164】

また、請求項2にかかる発明により、再生画像の表示に要する時間を大幅に短縮することが可能となる。

【0165】

請求項3に関わる発明によれば、時間軸方向にベクトル量子化を行うことにより、再生画像の質を保持したまま高い圧縮率で動画を圧縮することが出来る。

【0166】

請求項4に関わる発明によれば、時間軸方向にベクトル量子化を行うことにより、再生画像の質を保持したまま高い圧縮率で動画を圧縮することが出来る。

【0167】

請求項5に関わる発明によれば、時間軸方向にベクトル量子化を行うことにより、再生画像の質を保持したまま高い圧縮率で動画を圧縮することが出来る。

【0168】

請求項6に関わる発明によれば、時間軸方向にベクトル量子化を行うことにより、再生画像の質を保持したまま高い圧縮率で動画を圧縮することが出来る。

【0169】

請求項7に関わる発明によれば、高い圧縮率で動画を圧縮・転送することが出来る。

【0170】

請求項8に関わる発明によれば、高い圧縮率で動画を圧縮・転送することが出来る。

【0171】

請求項9に関わる発明によれば、コードブックの最適化をより効率的に行うことが出来る。

【0172】

請求項10に関わる発明によれば、コードブックの最適化をより効率的に行うことが出来る。

【0173】

請求項11に関わる発明によれば、コードブックの最適化をより効率的に行う

ことが出来る。

【0174】

請求項12に関わる発明によれば、コードブックの最適化をより効率的に行うことが出来る。

【0175】

請求項13に関わる発明によれば、ベクトル量子化を行う際に必要な演算量を減少させることが出来、演算を高速に行うことが出来る。

【0176】

請求項14に関わる発明によれば、ベクトル量子化を行う際に必要な演算量を減少させることが出来、演算を高速に行うことが出来る。

【0177】

請求項15に関わる発明によれば、ベクトル量子化を行う際に必要な演算量を減少させることが出来、演算を高速に行うことが出来る。

【0178】

請求項16に関わる発明によれば、ベクトル量子化により圧縮伸長を行う際に必ず必要となるコードブックを別に用意することにより、実際のデータのみを転送するだけでよく、より容量の少ない通信媒体で転送することが可能となる。

【図面の簡単な説明】

【図1】

ベクトル量子化PCIバスボードの一例を示す写真である。

【図2】

ベクトル量子化システム全体のデータの流れを説明するための模式図である。

【図3】

ベクトル量子化PCIバスボードによりベクトル量子化をさせた時に得られた動画像の1フレームの写真である。

【図4】

640×480画素の1枚の画像を処理するために要する時間を測定した結果を示すグラフである。

【図5】

ベクトルとして画像上の走査線に沿ったラインブロックをベクトルデータとして用いた方式の特徴を示す模式図である。

【図 6】

ラインブロックをベクトルデータとして用いた場合のデータを貼り込む位置を説明する模式図である。

【図 7】

図 6 のスライド配置方式により画像を圧縮伸長した結果得られた再生画像である。

【図 8】

時間軸方向のベクトル量子化の概念を説明する概念図である。

【図 9】

時間軸方向のベクトル量子化の概念を説明する概念図である。

【図 10】

時間軸方向のベクトル量子化の概念を説明する概念図である。

【図 11】

コードブック方式動画像圧縮装置の構成の一例を示す概念図である。

【図 12】

コードブック方式動画像圧縮処理をソフトウェアで実現する場合のコードブック方式動画像圧縮プロセッサの構成を説明する概念図である。

【図 13】

コードブック方式動画像圧縮処理の手順の一例を示すフロー・チャートである。

【図 14】

初期のコードブックとして輝度値が連続的に変化したもの（とり得る値の最小値から最大値まで連続的に変化するもの）と、輝度値 128 で一定のものと、ランダムパターンのときに、このコードブックを用いて画像を圧縮・伸長したときの P S N R 特性の違いを説明するグラフである。

【図 15】

マップ上（仮想の 2 次元平面）での、書き換えを行う範囲を 1 次元的に適用さ

せ更新回数の増加とともに範囲を減少させていく方法と2次元的に適用させ更新回数の増加とともに範囲を減少させていく方法のPSNR特性の違いを説明するグラフである。

【図16】

利得を更新回数の増加にかかわらず初期値から減少させないときと更新回数の増加とともに初期値から減少させるときのPSNR特性の違いを説明するグラフである。

【図17】

更新回数160000回のときの利得とPSNR特性の関係を表したグラフである。

【図18】

最適化で得られた3種類のコードブックAとBとCの特徴を一つのコードブック $A+B+C$ にまとめる手法を示したものである。

【図19】

A、B、C、 $A+B+C$ それぞれのコードブックを用いて画像A、B、Cに対してベクトル量子化した後復元した画像のPSNRを示すグラフである。

【図20】

2つのベクトルが類似しているか否かを数値化した類似度を求める関数としてマンハッタン距離を、特徴量としてベクトルデータの要素の総和を利用した場合のデータ圧縮システムにおけるベクトル量子化の装置の構成と、演算が省略される様子とを説明する概念図である。

【図21】

ベクトル量子化を用いて画像圧縮を行う例を示す模式図である。

【図22】

画像データに対してベクトル量子化を行ったときに、マンハッタン距離演算がどれほど省略出来るかを示すグラフである。

【図23】

画像データに対して、類似度を演算する関数をマンハッタン距離、特徴量としてベクトル要素の総和とした場合に問題が生じる点を示したものである。

【図 24】

ベクトルデータの一部を反転させた場合、画像ブロックがどのように変化するかをあらわす模式図である。

【図 25】

ベクトルデータ要素の一部を画素の明暗が反転するように操作した後、ベクトルデータの要素の総和を求めて特徴量としたとき、特徴量の差が現れる例を説明する模式図である。

【図 26】

画像ブロックの四隅を特徴量として用いる例を示す模式図である。

【図 27】

画像の変化の周期が同じである 2 つの画像ブロックを示す模式図である。

【図 28】

データ圧縮システムにおけるベクトル量子化装置の構成を説明する概念図である。

【図 29】

コードブック・サーバシステムとその運用構成を説明する概念図である。

【符号の説明】

- 601 画像入力装置、
- 602 コードブック記憶装置、
- 603 コードブックによる圧縮装置、
- 604 コード列記憶装置、
- 605 コード列比較装置、
- 606 転送媒体、
- 7-101 CPU、
- 7-102 メモリ、
- 7-103 コードブック記憶エリア、
- 7-104 コード列記憶エリア、
- 7-105 ポインタ・カウンタ・エリア、
- 7-106 転送閾値記憶エリア、

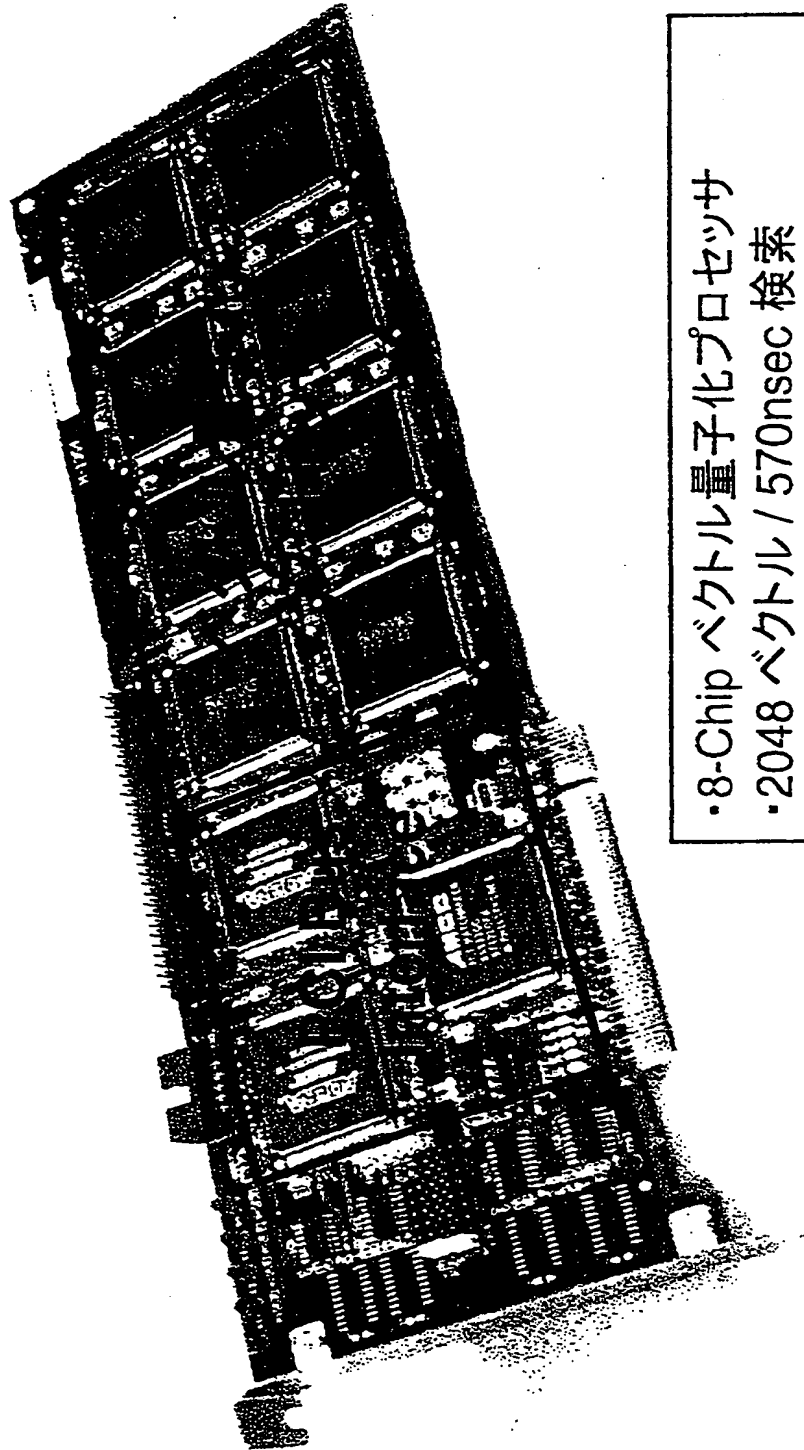
- 1201 テンプレート記憶装置、
- 1202 マンハッタン距離演算装置、
- 1203 最小マンハッタン距離記憶装置、
- 1204 特徴量記憶装置、
- 1205 特徴量演算装置、
- 1206 特徴量差分演算装置、
- 1207 演算省略判定装置、
- 1210 入力ベクトルデータ、
- 1211~1214 テンプレートベクトルデータ、
- 1215~1218 特徴量。



【書類名】

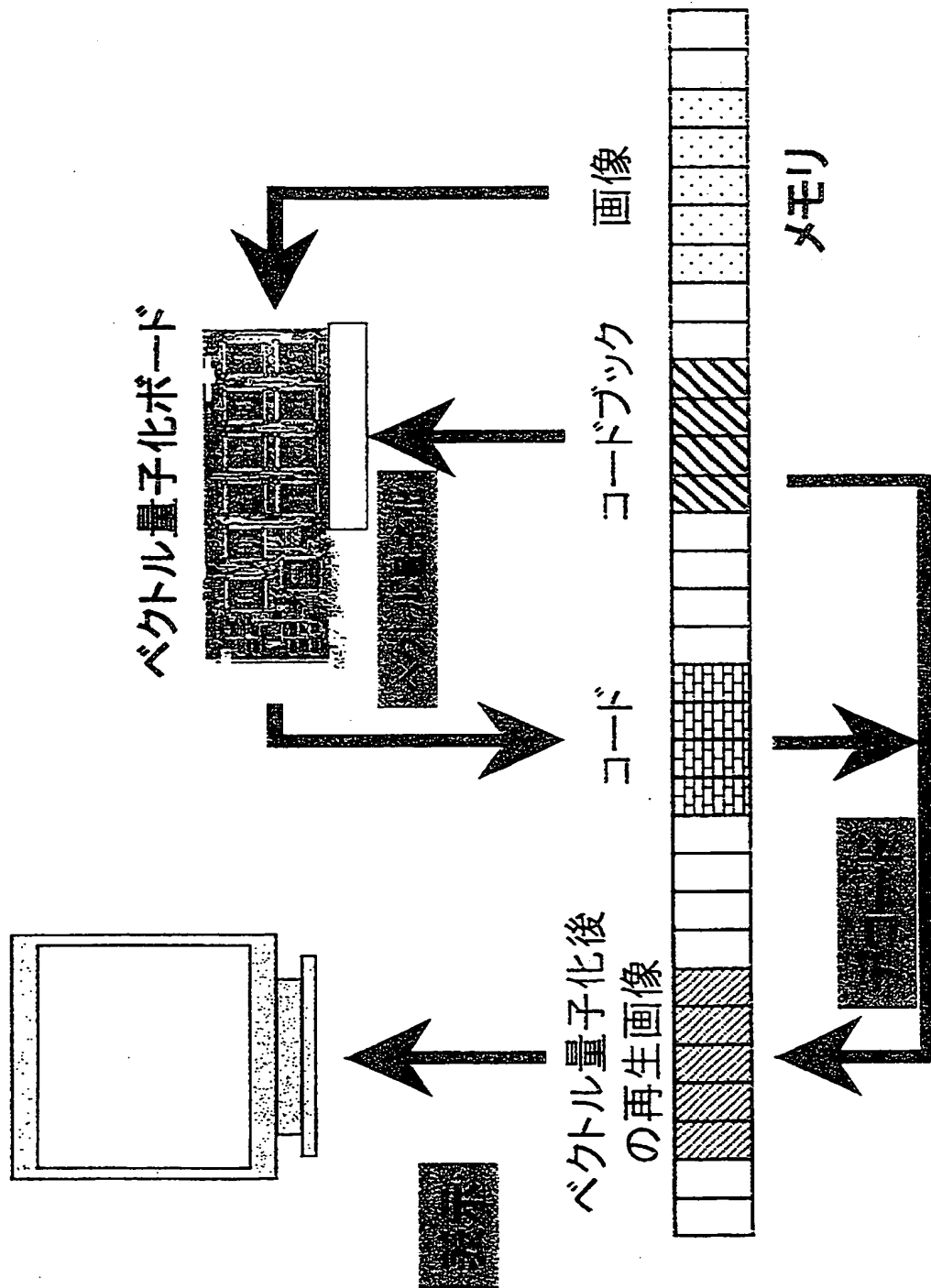
図面

【図1】



・8-Chip ベクトル量子化プロセッサ  
・2048 ベクトル / 570nsec 検索

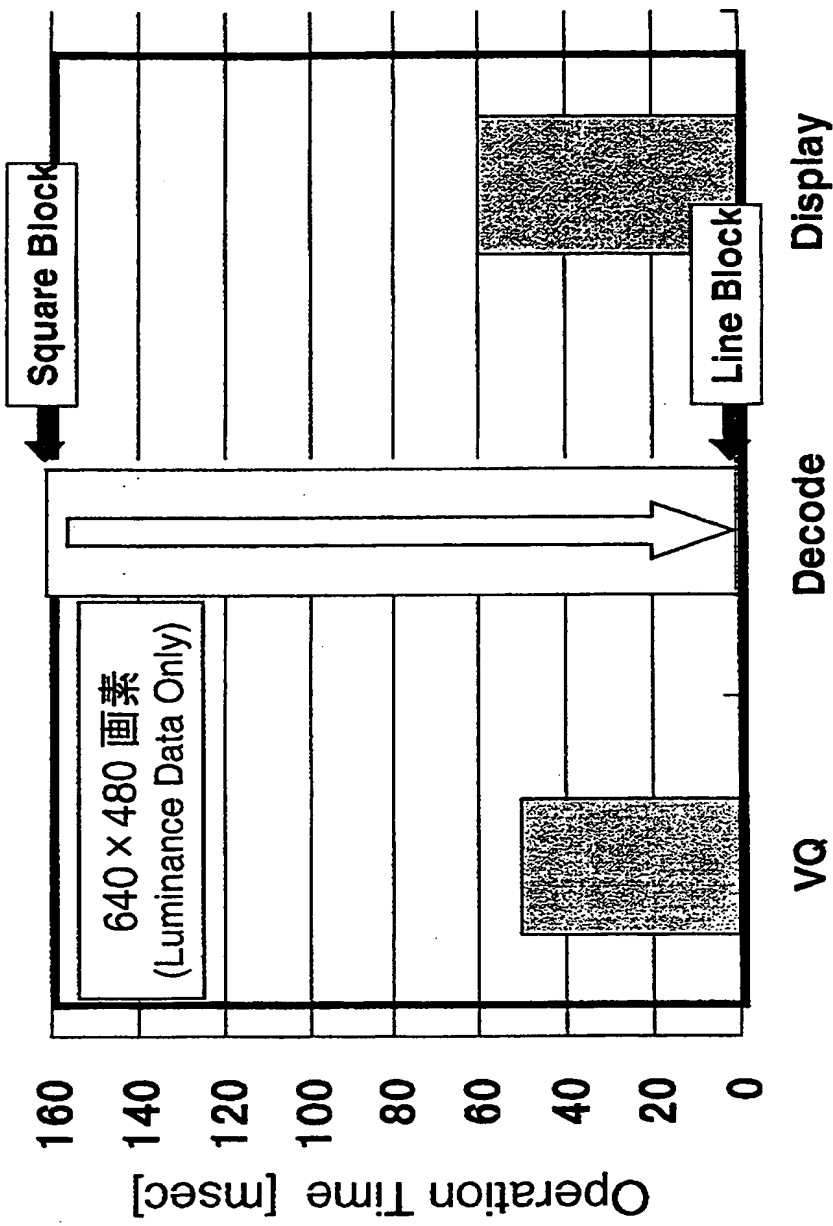
【図 2】



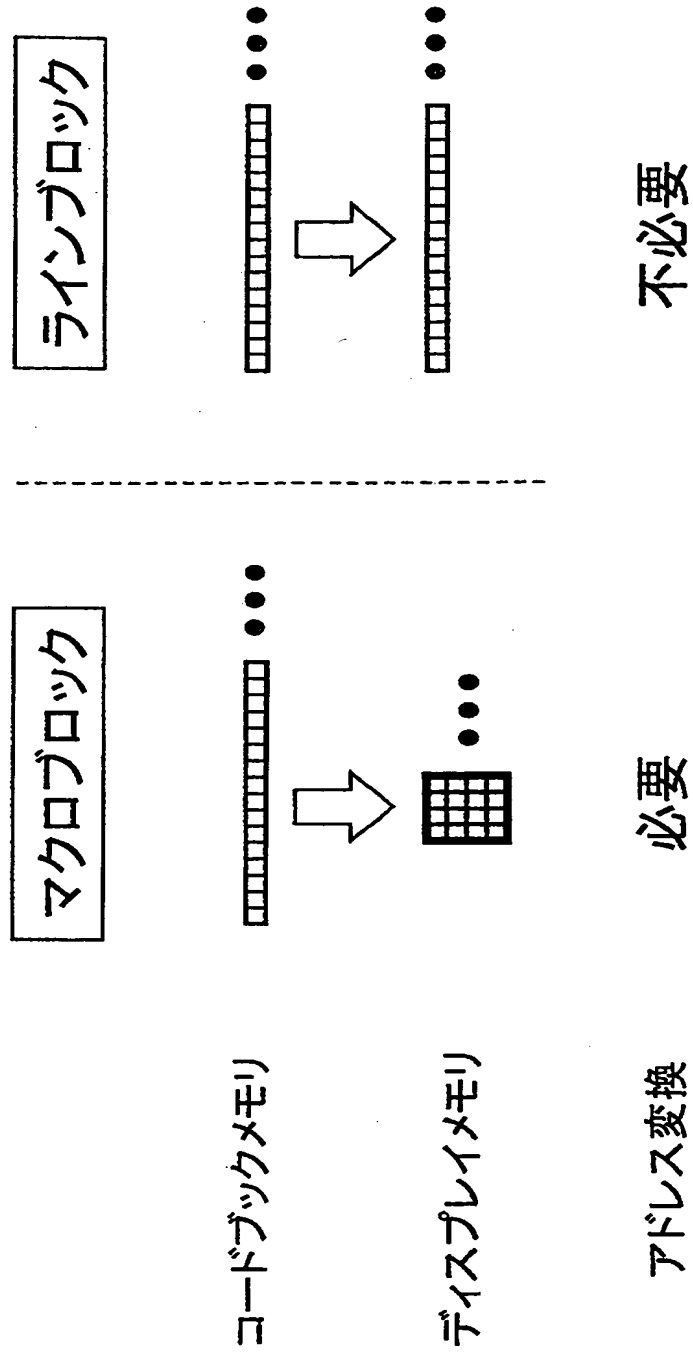
【図3】



【図 4】



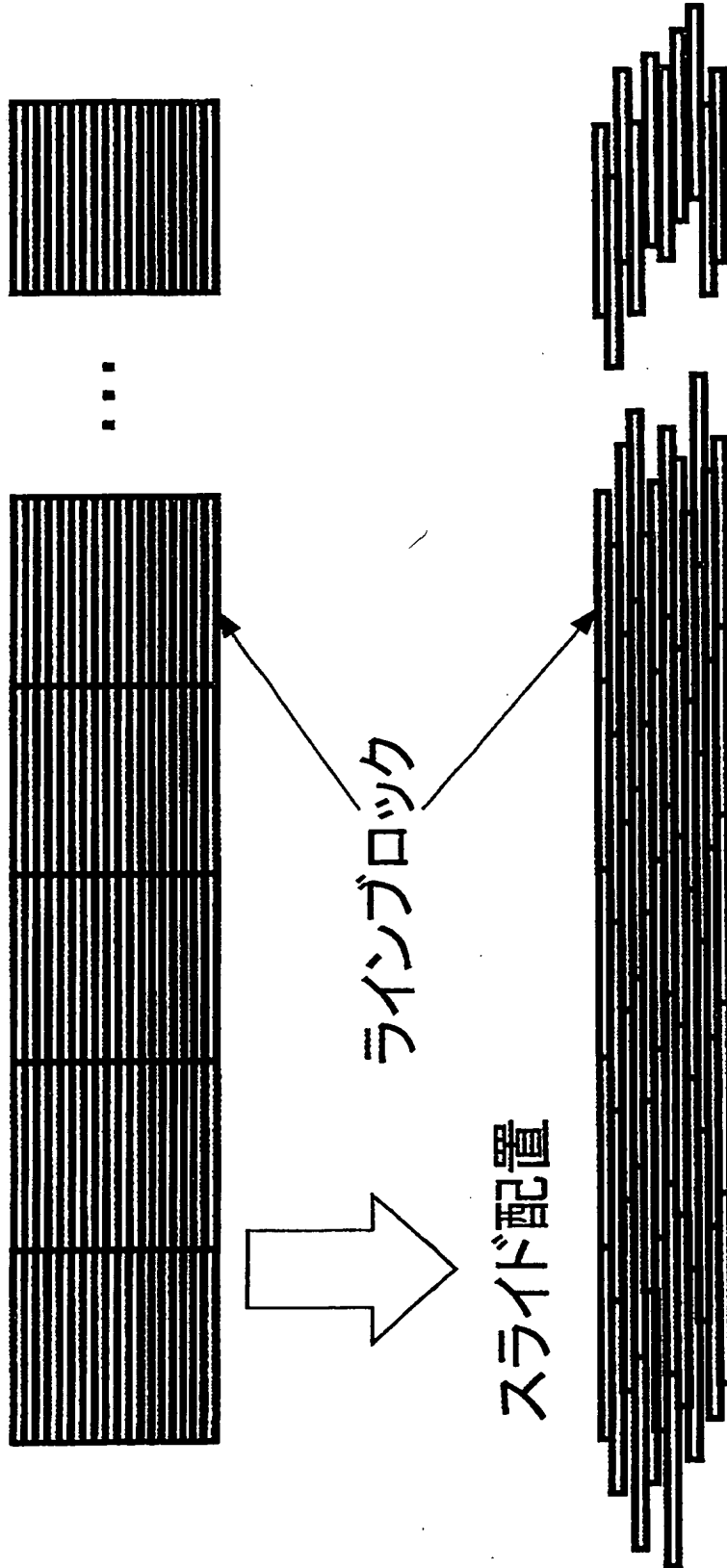
【図 5】



特平 1 0 — 1 2 4 2 8 8

【図 6】

通常配置

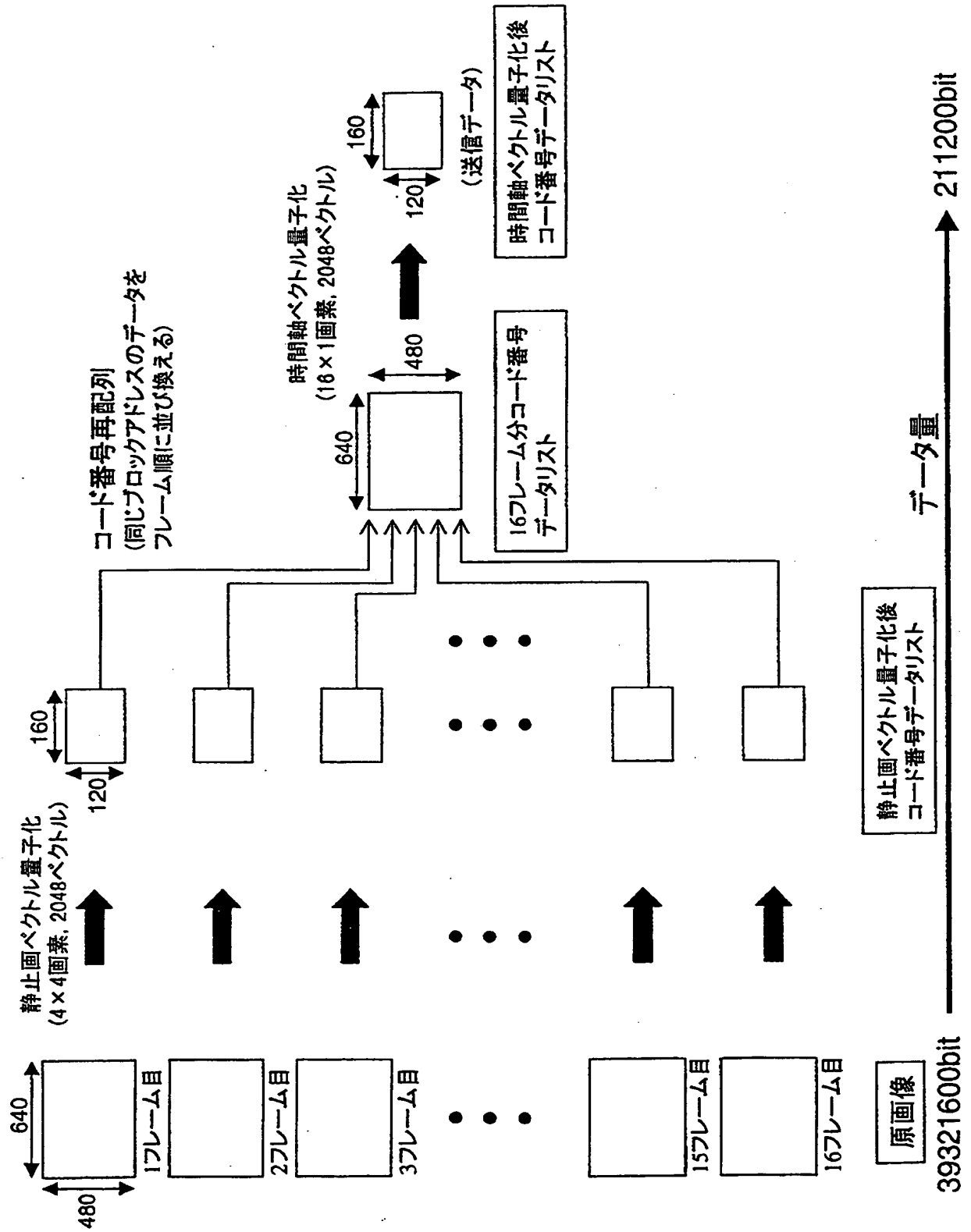


【図7】



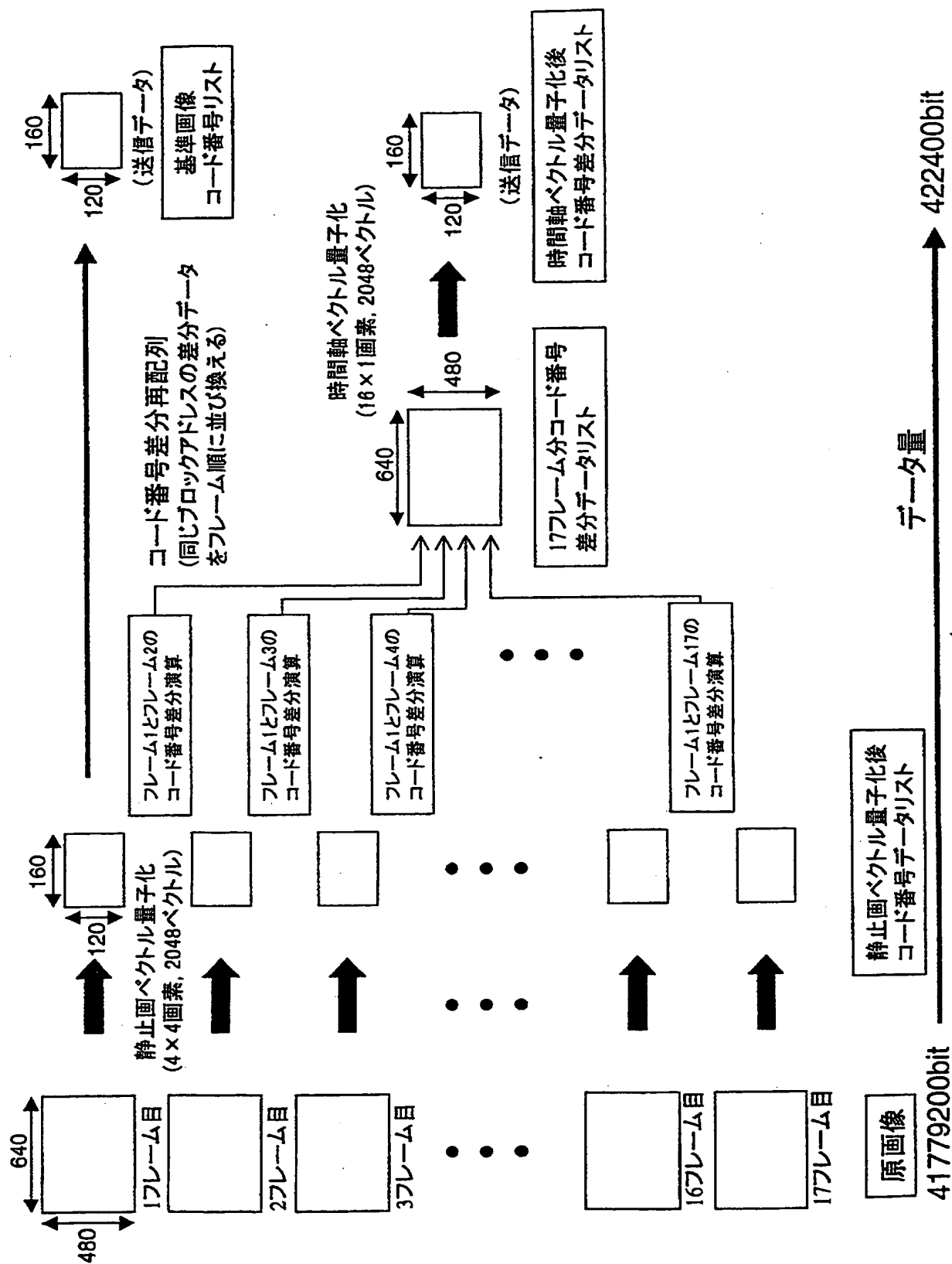


【図 8】

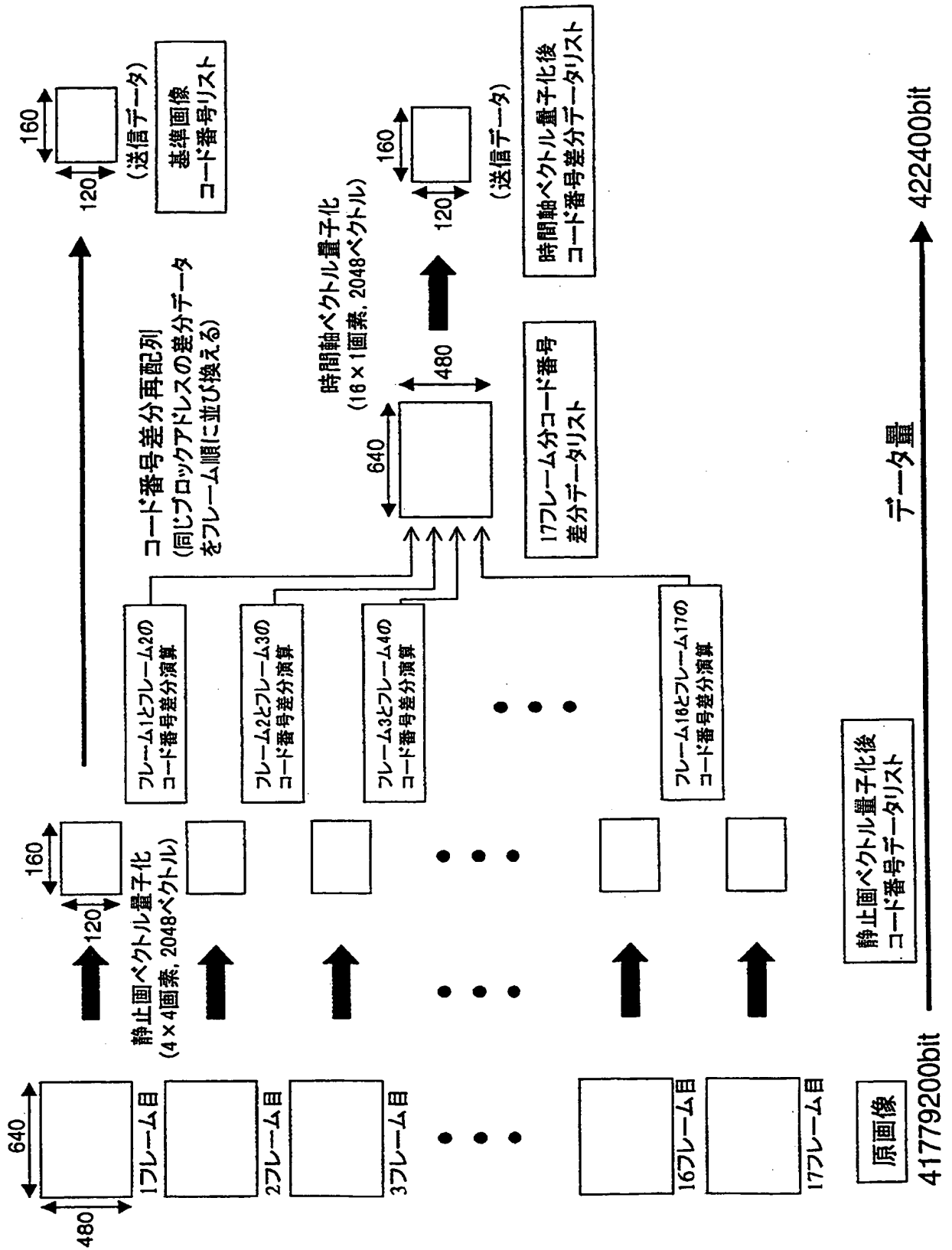


特平 1 0 - 1 2 4 2 8 9

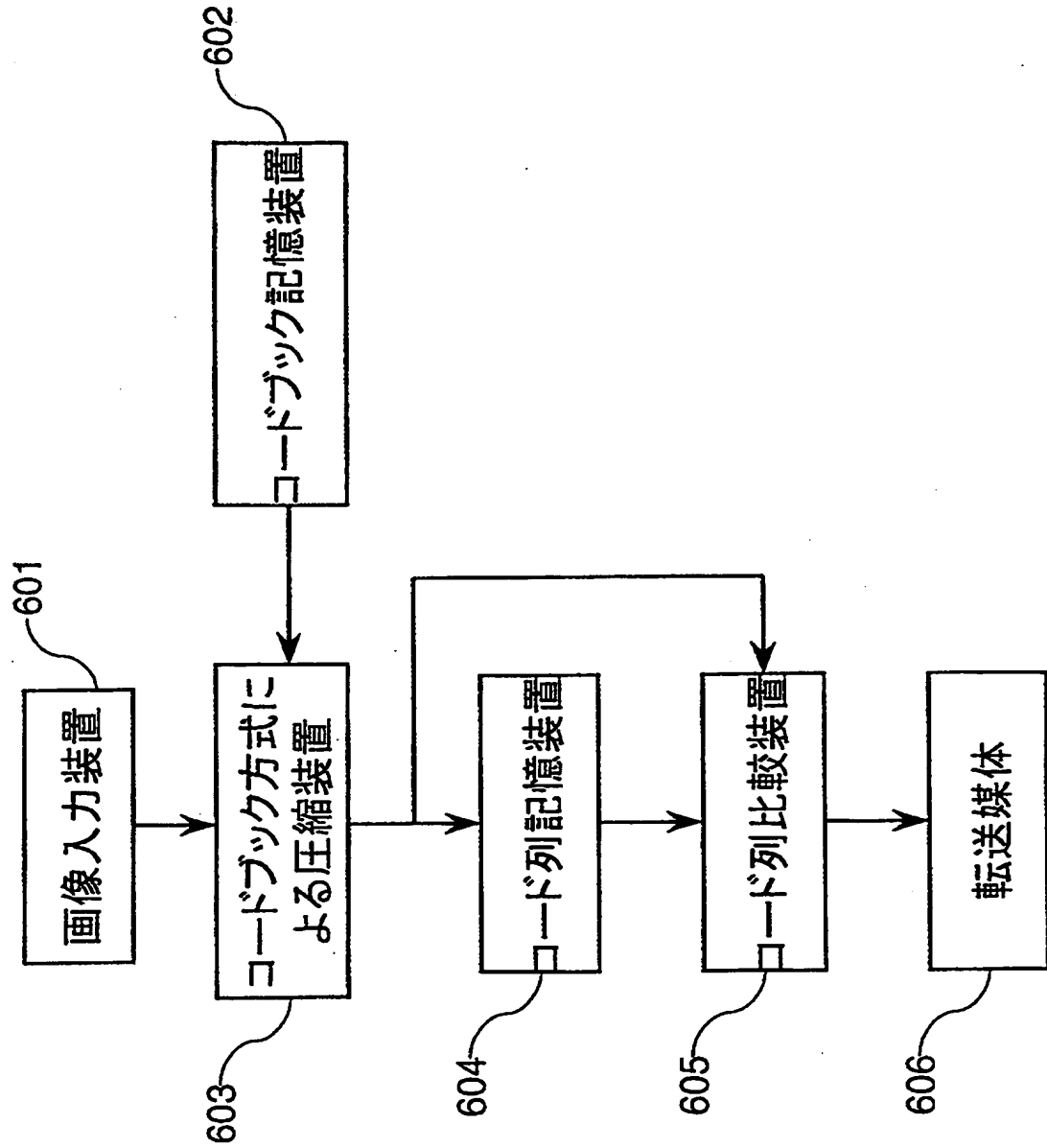
【図 9】



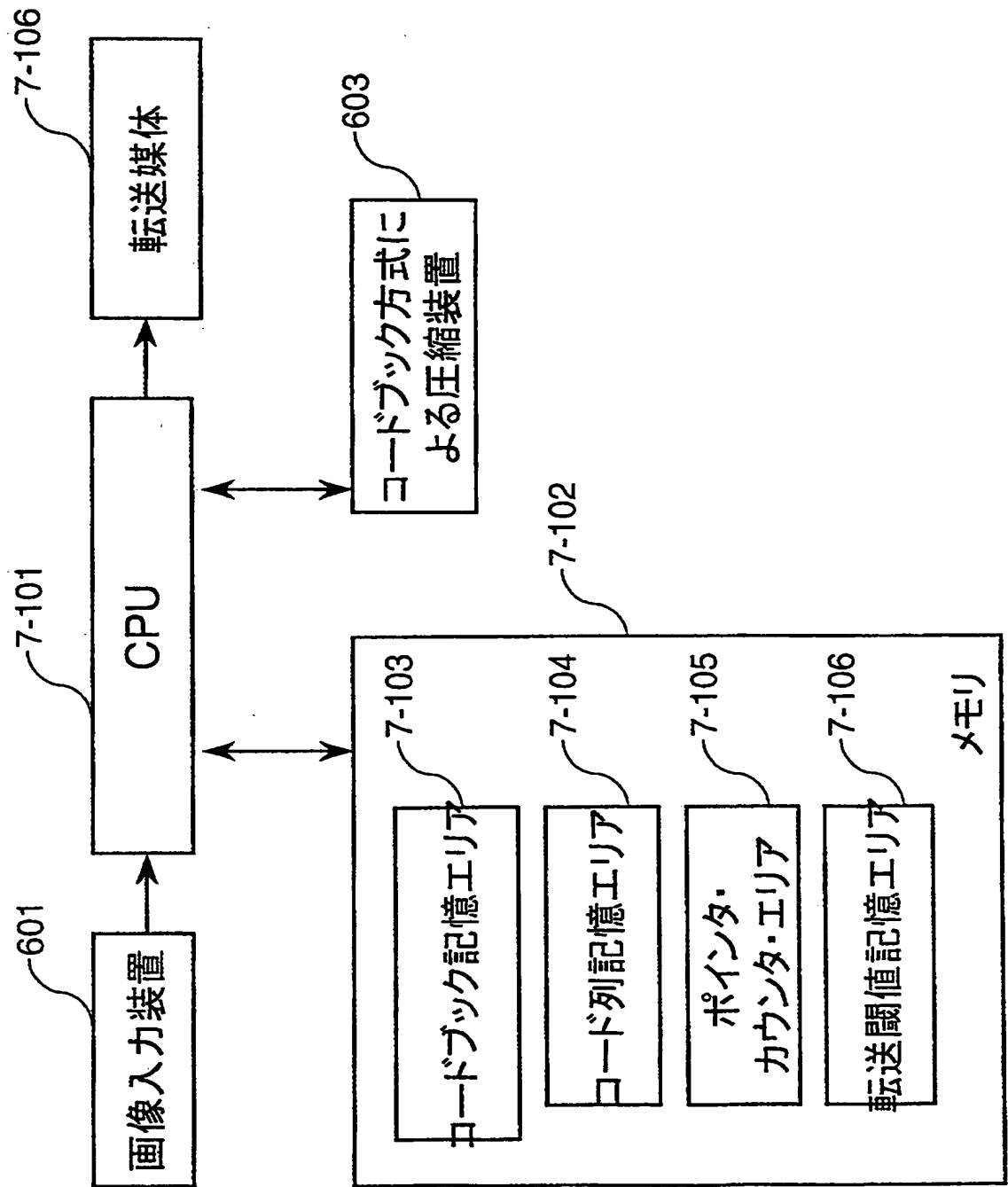
【図 10】



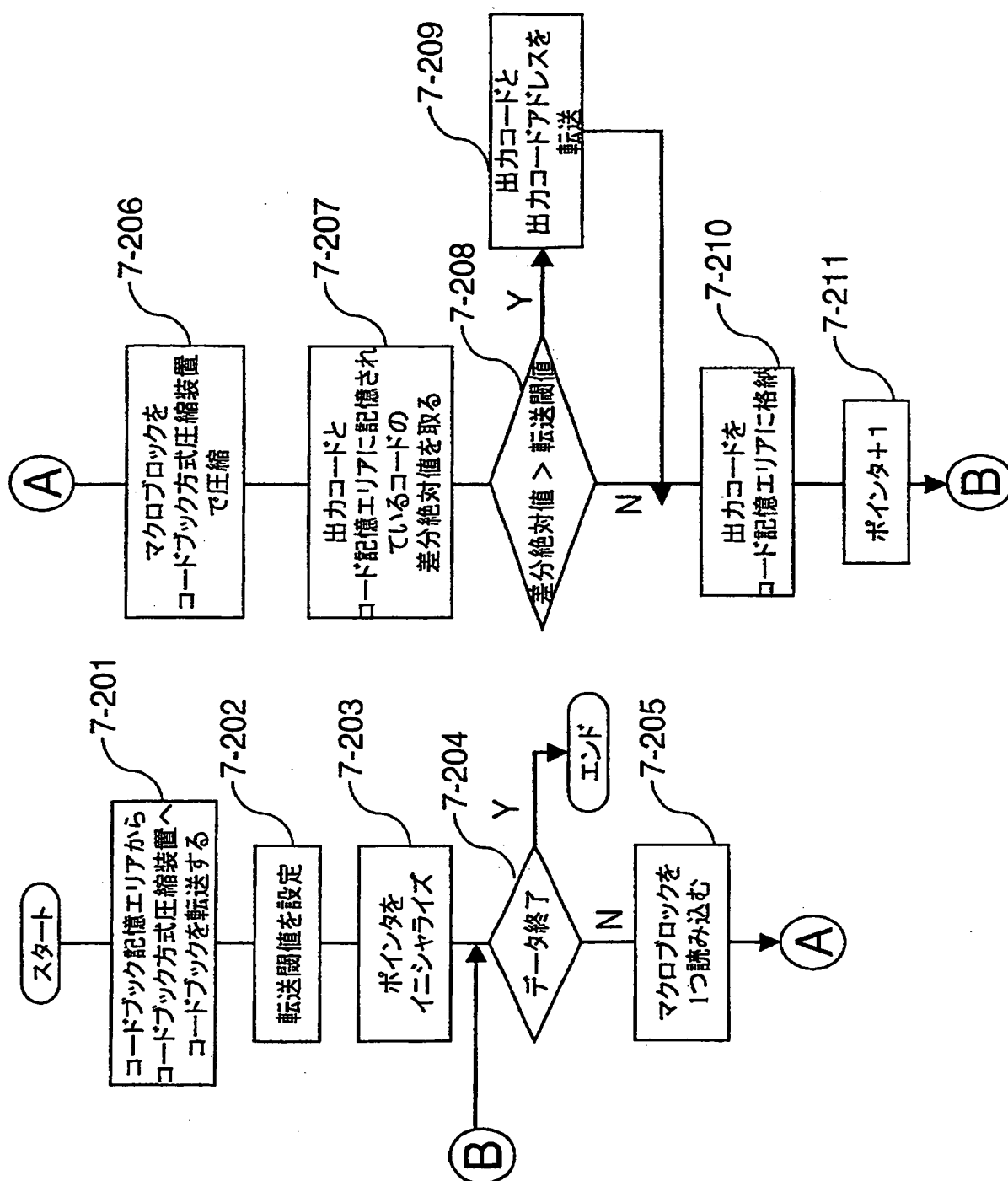
【図 11】



【図 12】



【图 13】

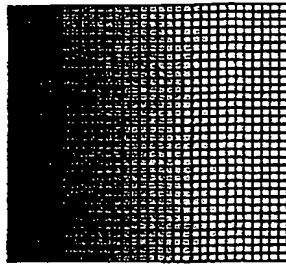


特平 1 0 - 1 2 4 2 8 0

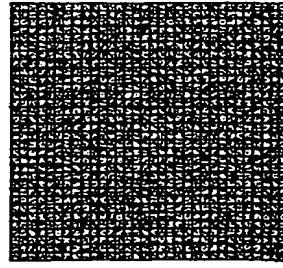
【図 1 4】



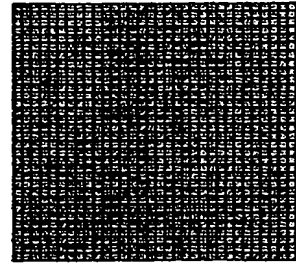
初期コードブックの種類



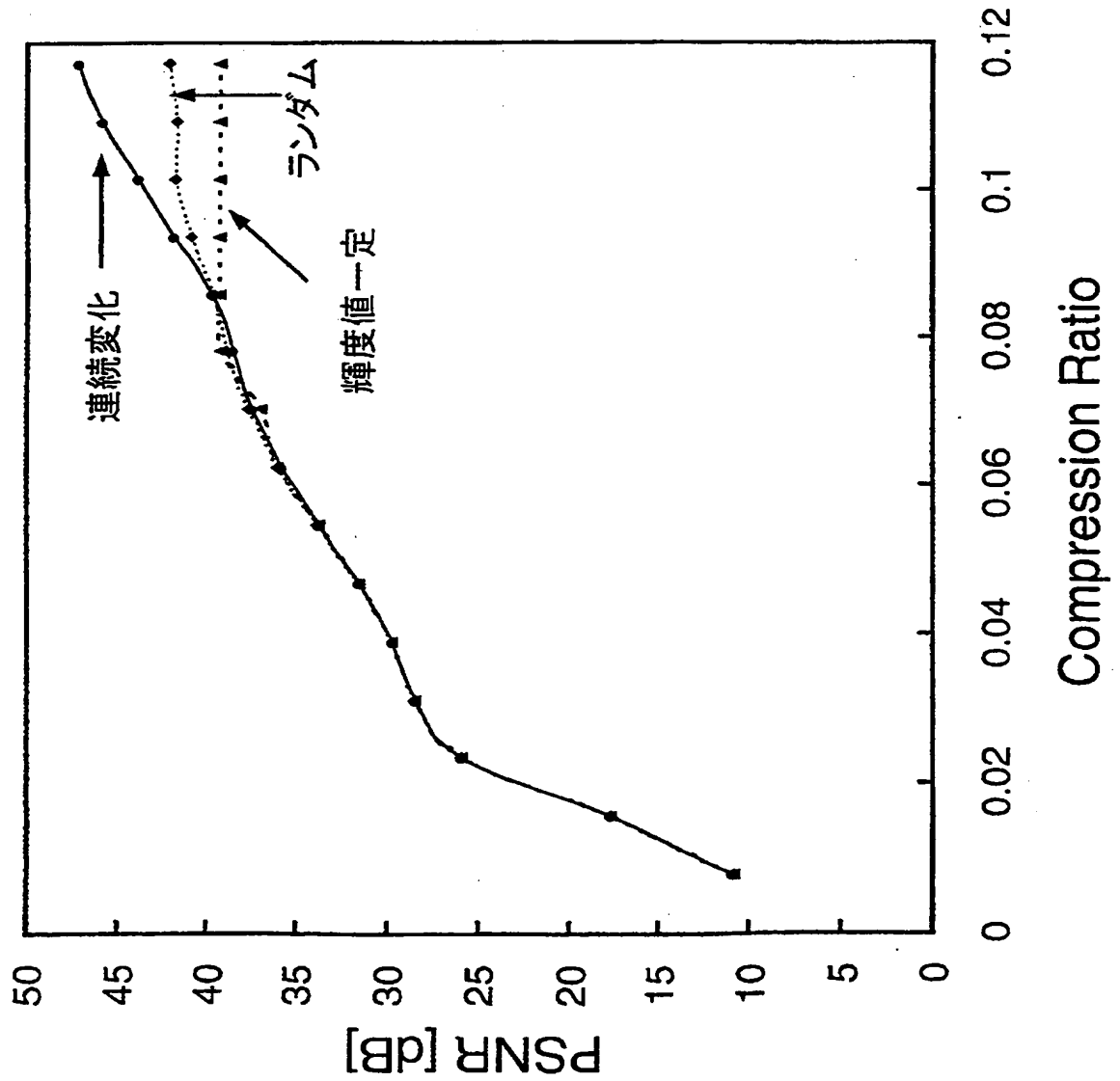
連続変化



ランダム



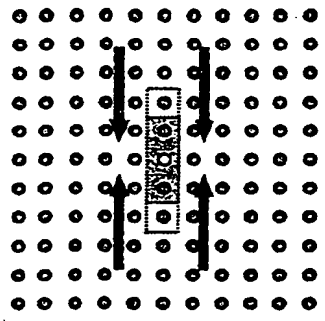
輝度値一定



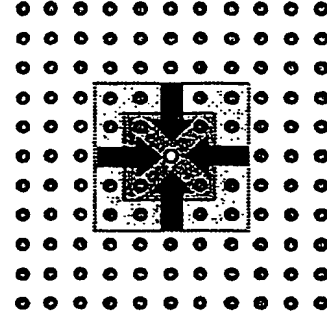
特平 10-124286

【図 15】

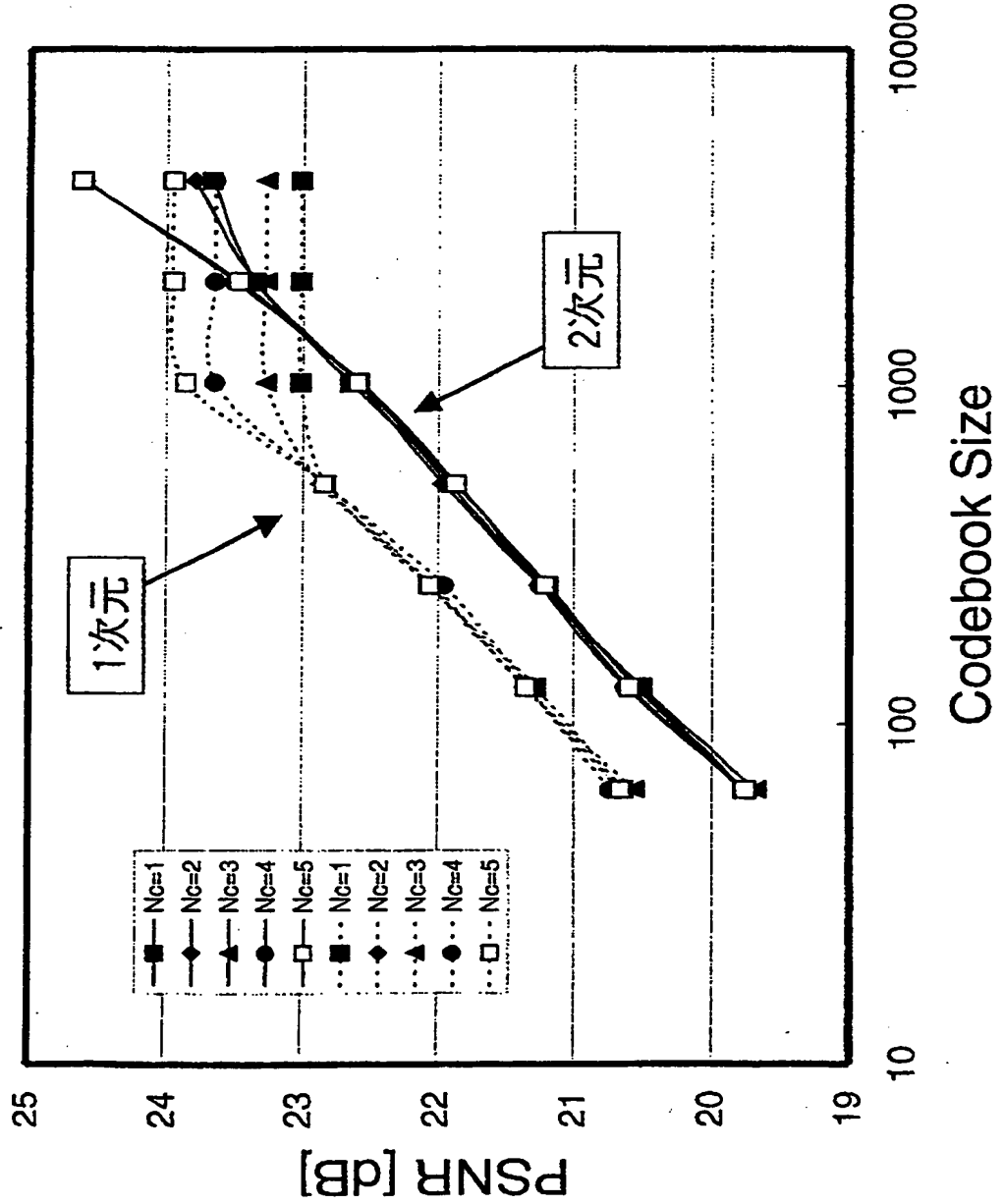
書き換えを行う範囲



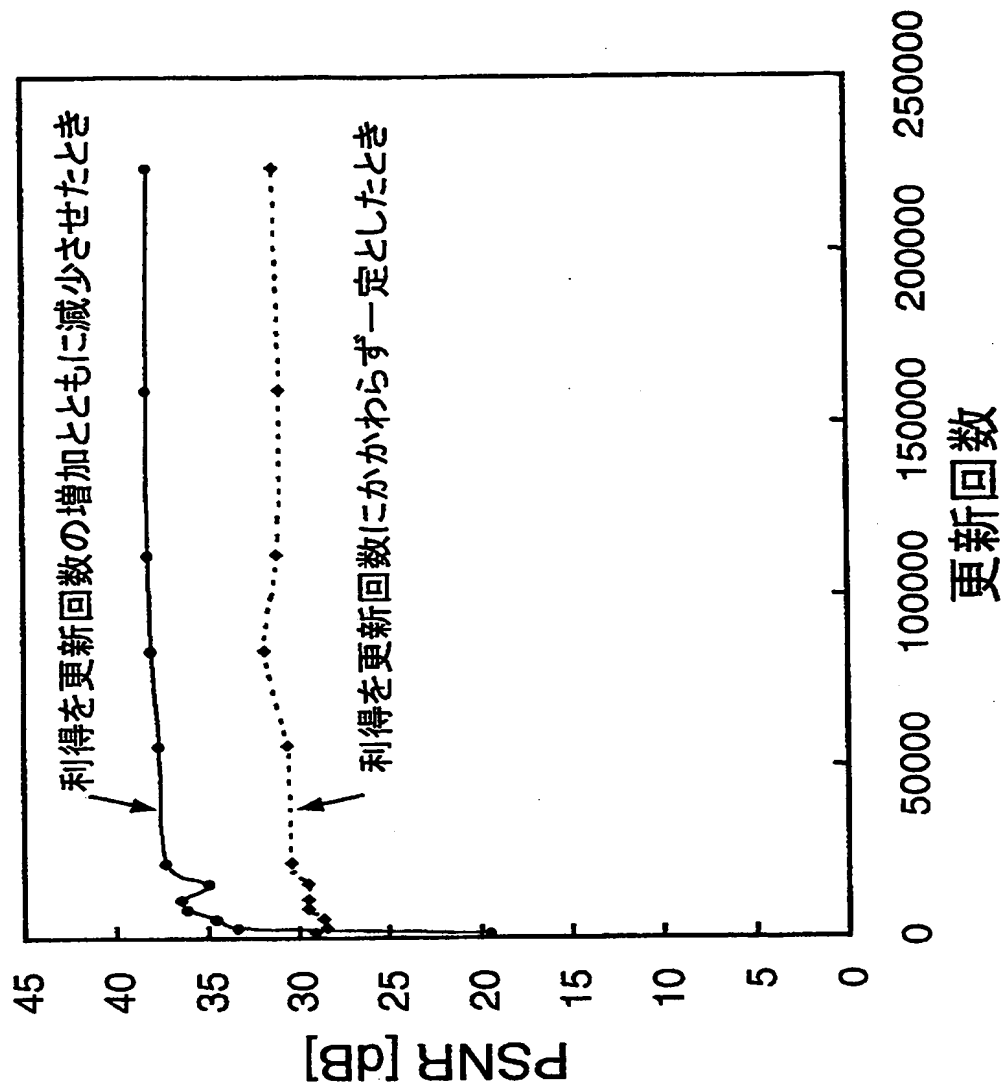
1次元的に適用させ  
更新回数が増加とともに  
範囲を減少させる方法



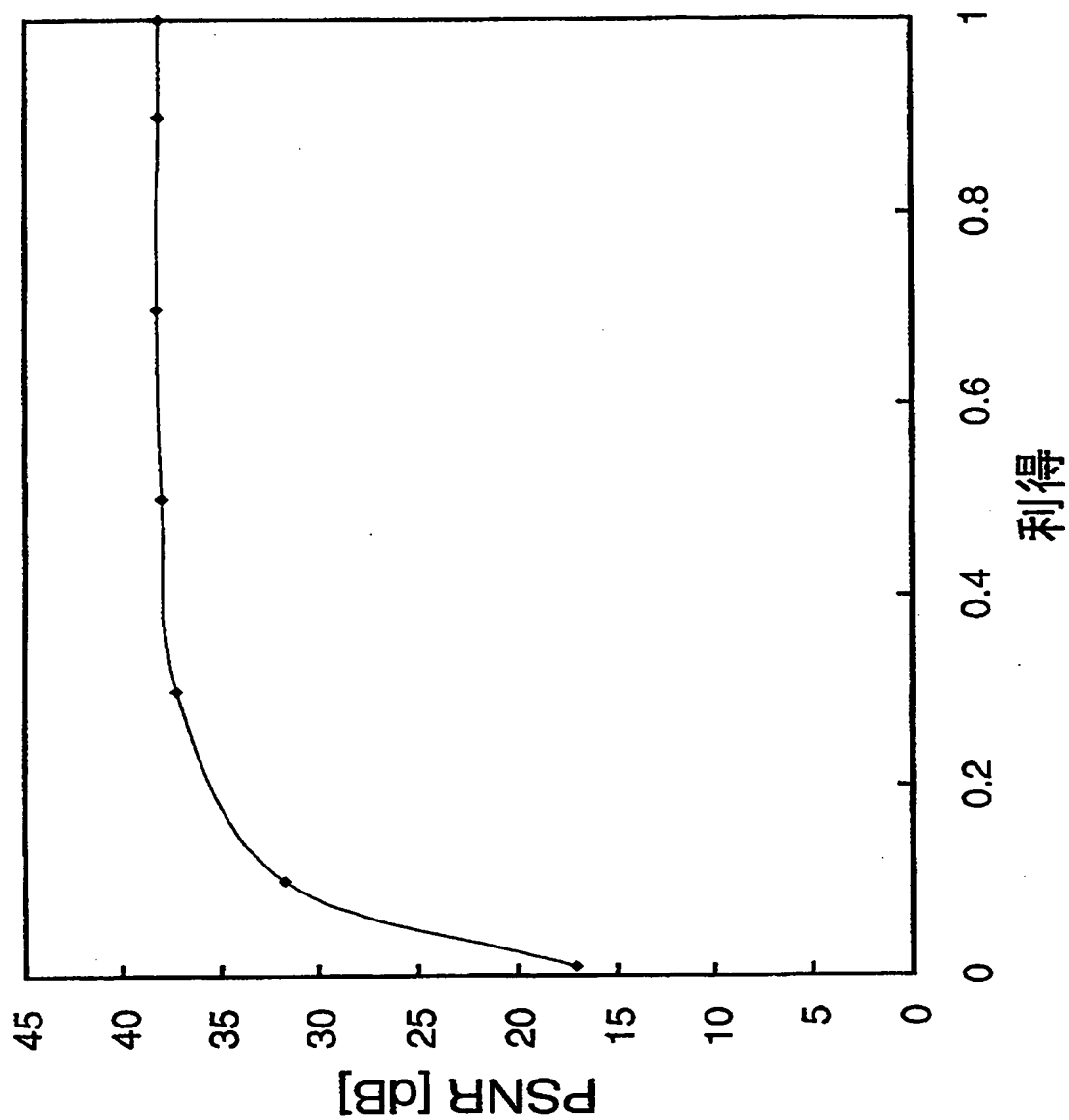
2次元的に適用させ  
更新回数が増加とともに  
範囲を減少させる方法



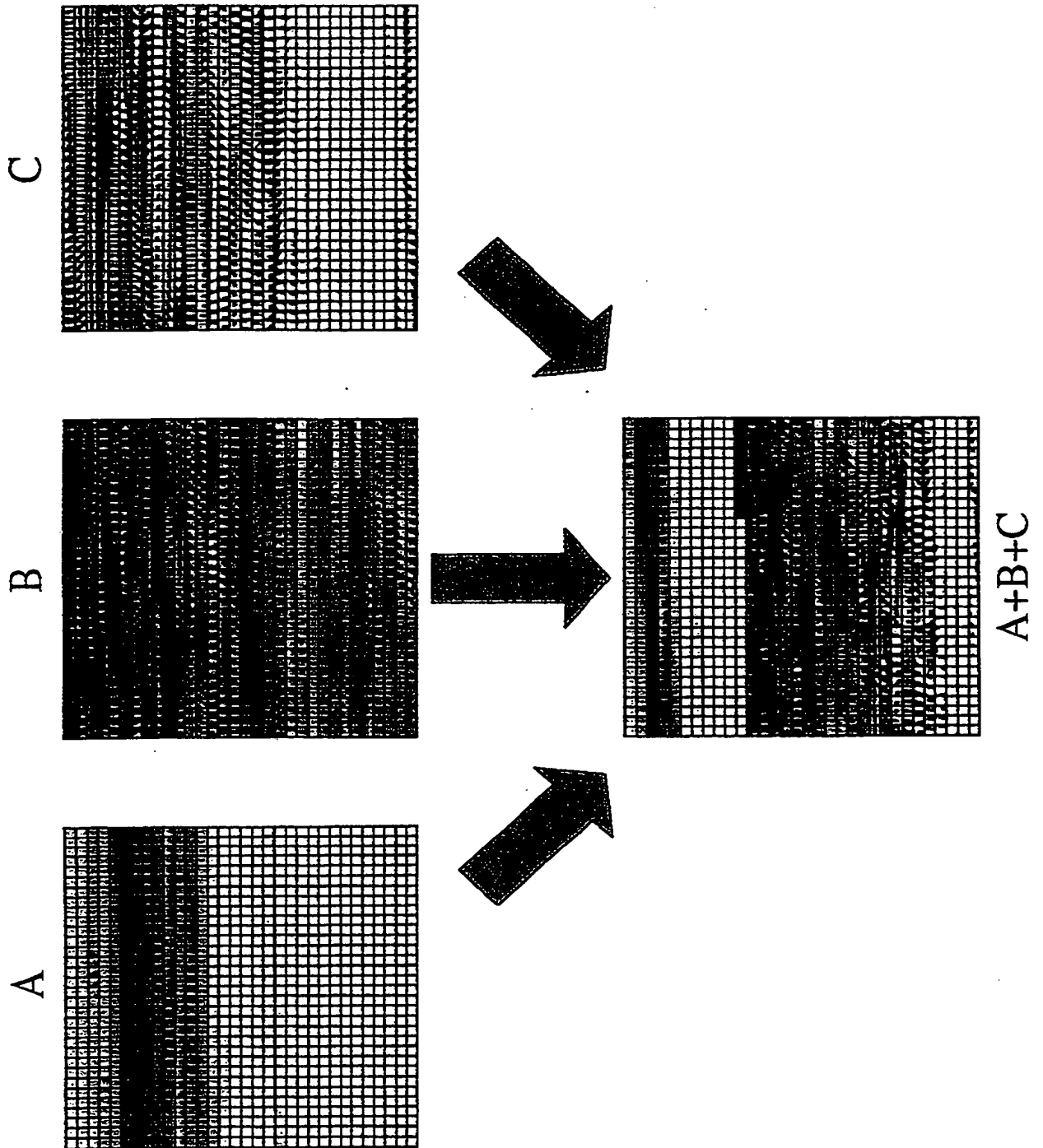
【図 16】



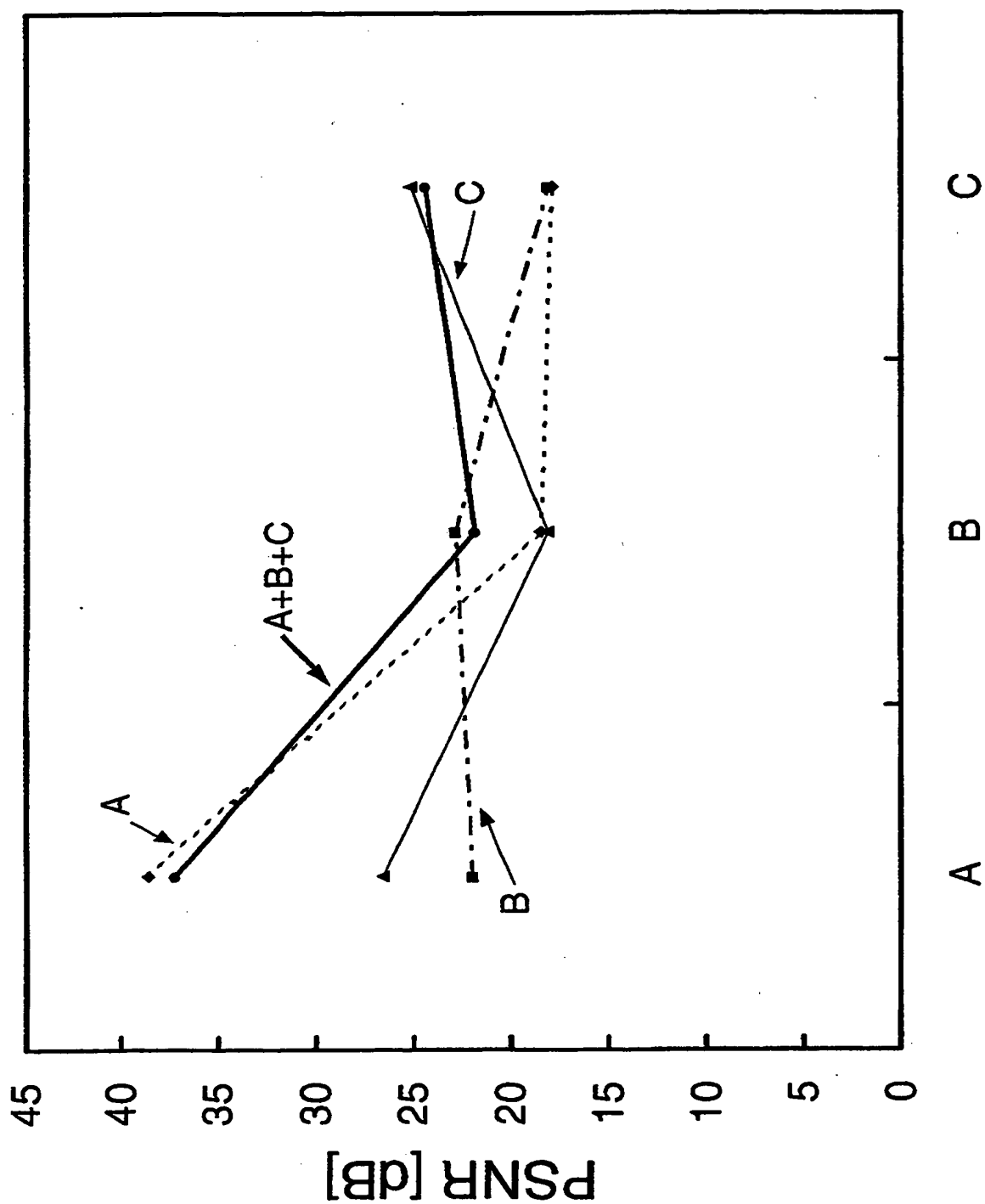
【図 17】



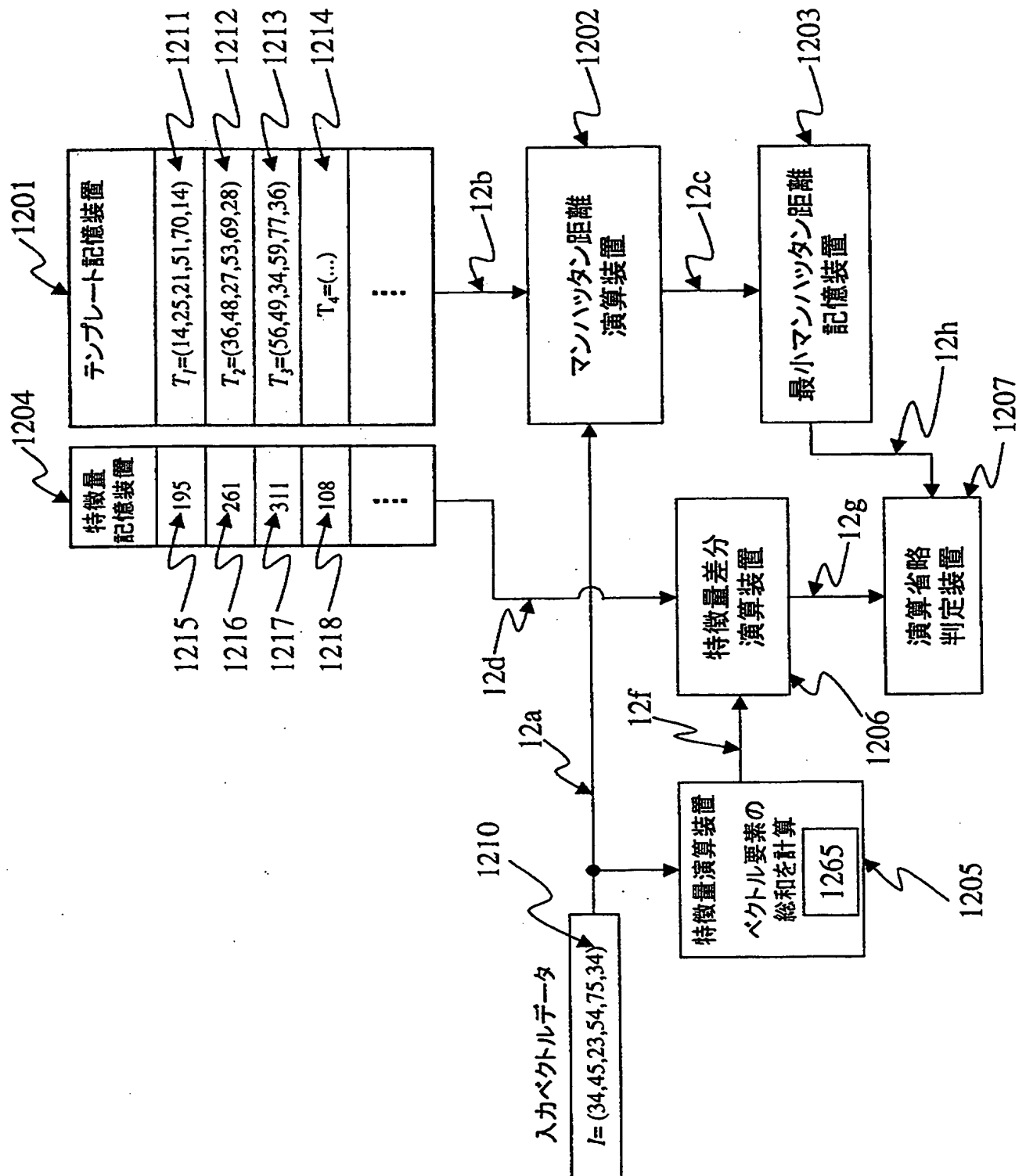
【図18】



【図19】

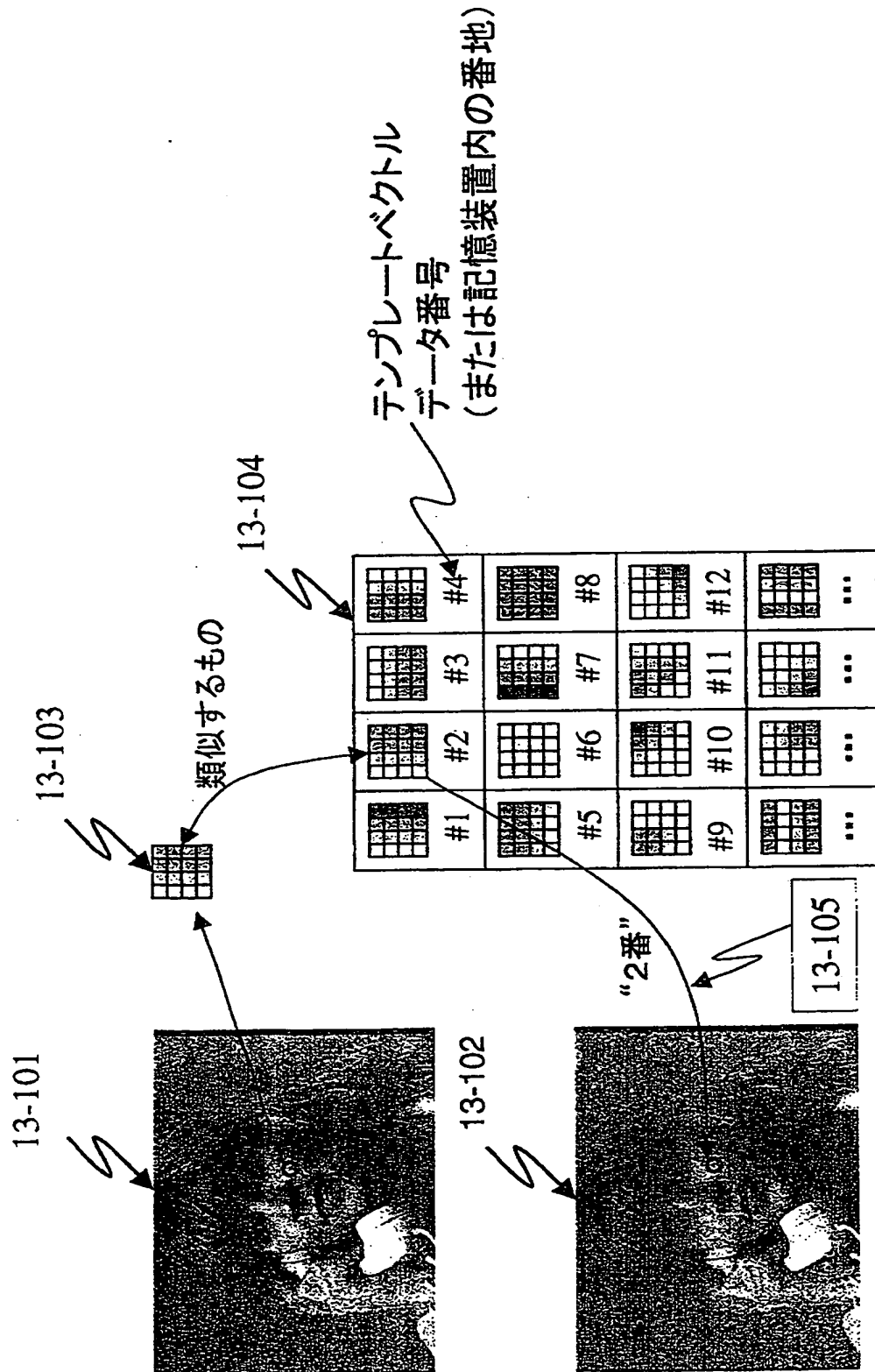


【図 20】





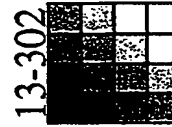
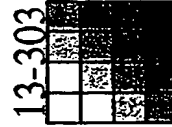
【図 21】





【図 2 3】

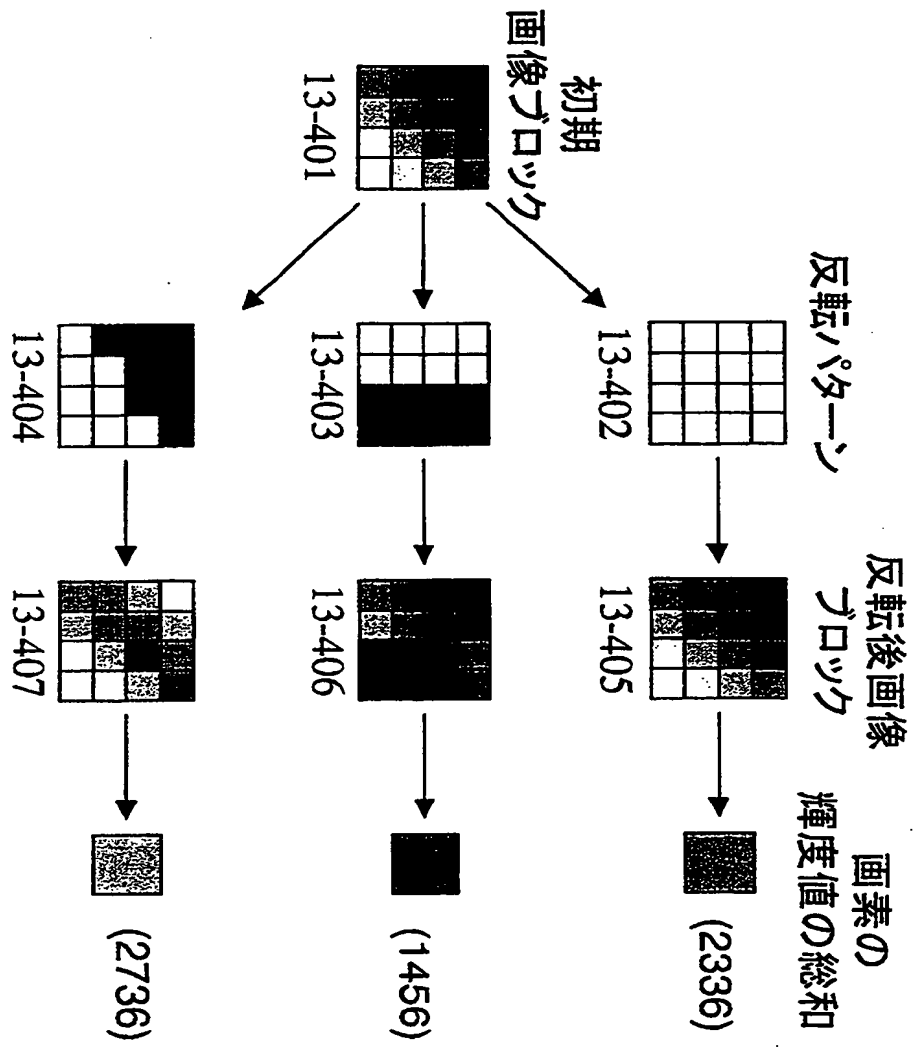
テンプレートベクトルデータに  
対応する画像ブロック



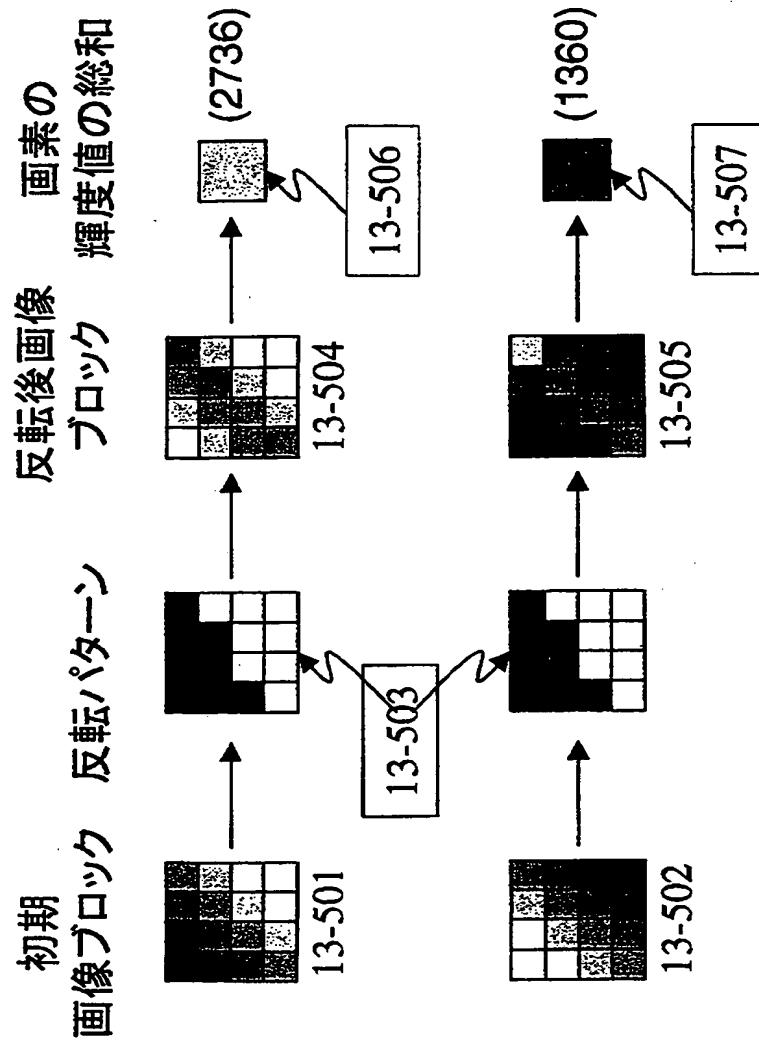
入力ベクトルデータに  
対応する画像ブロック



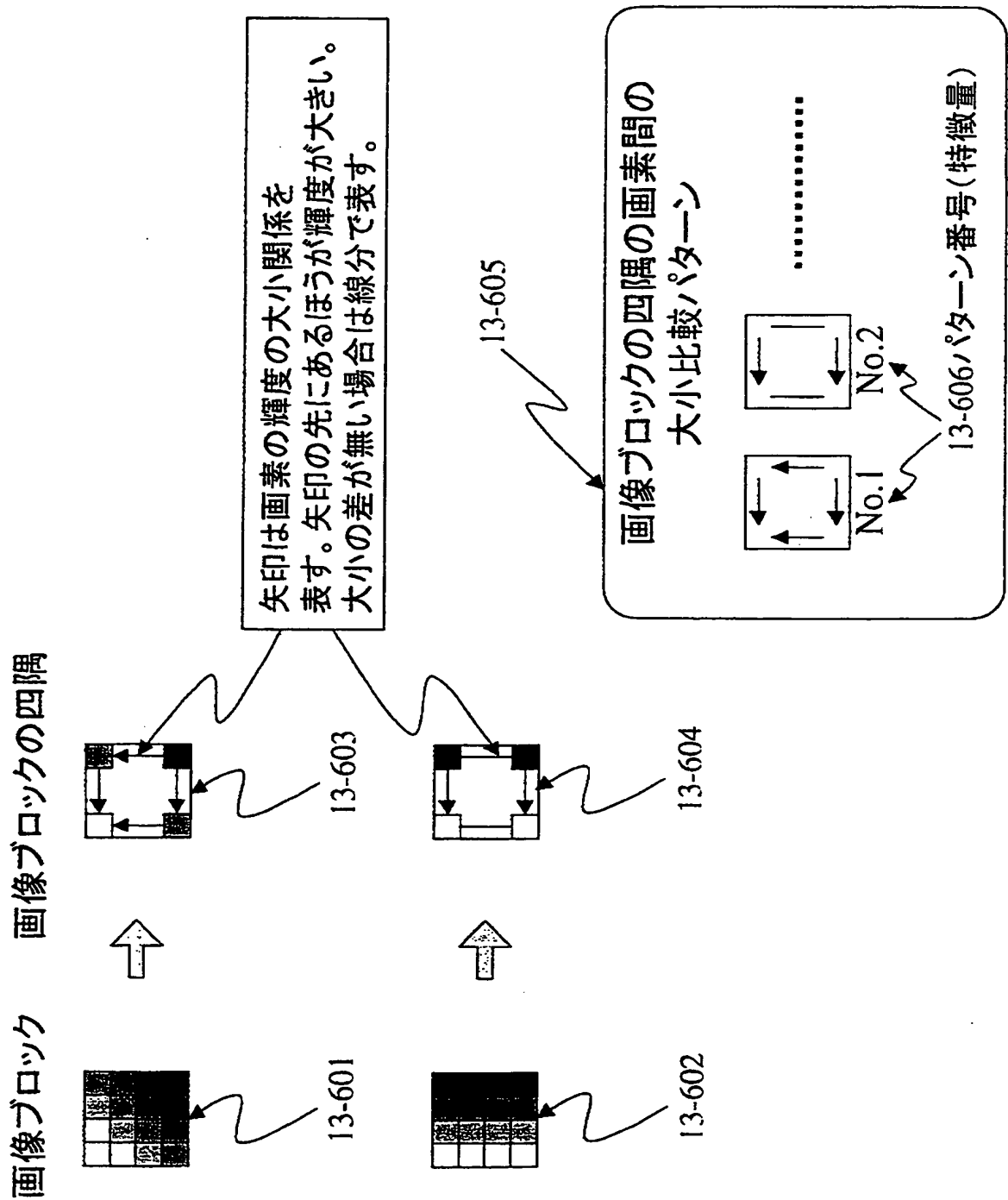
【図 24】



【図 25】

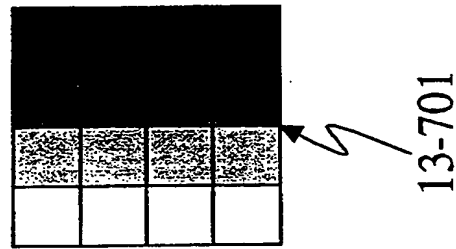
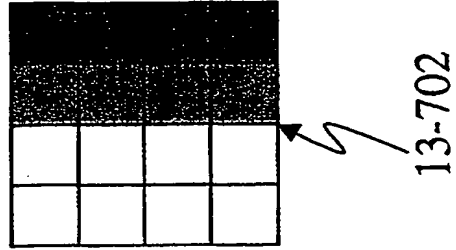


【図 26】

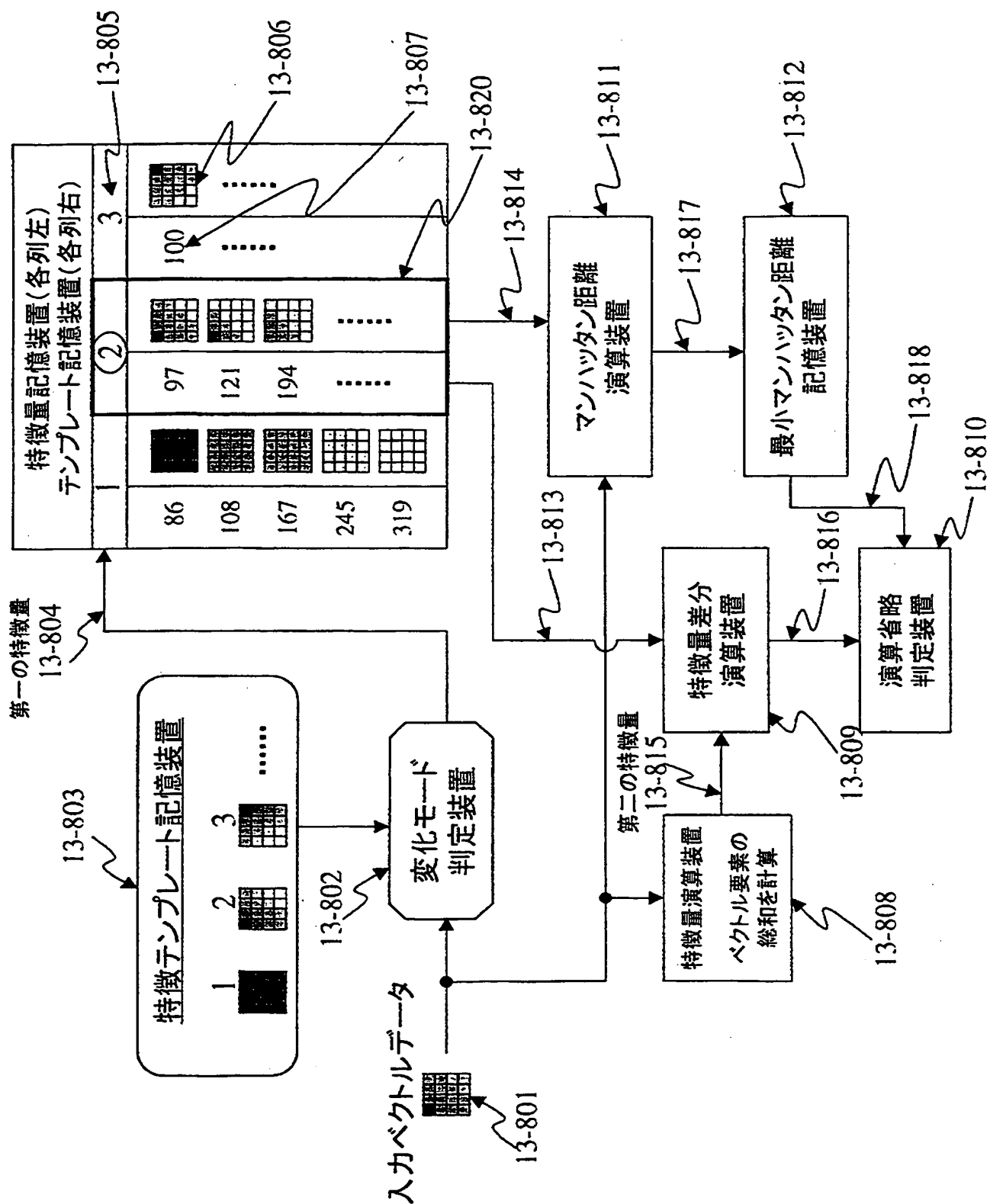


【図 27】

画像ブロック

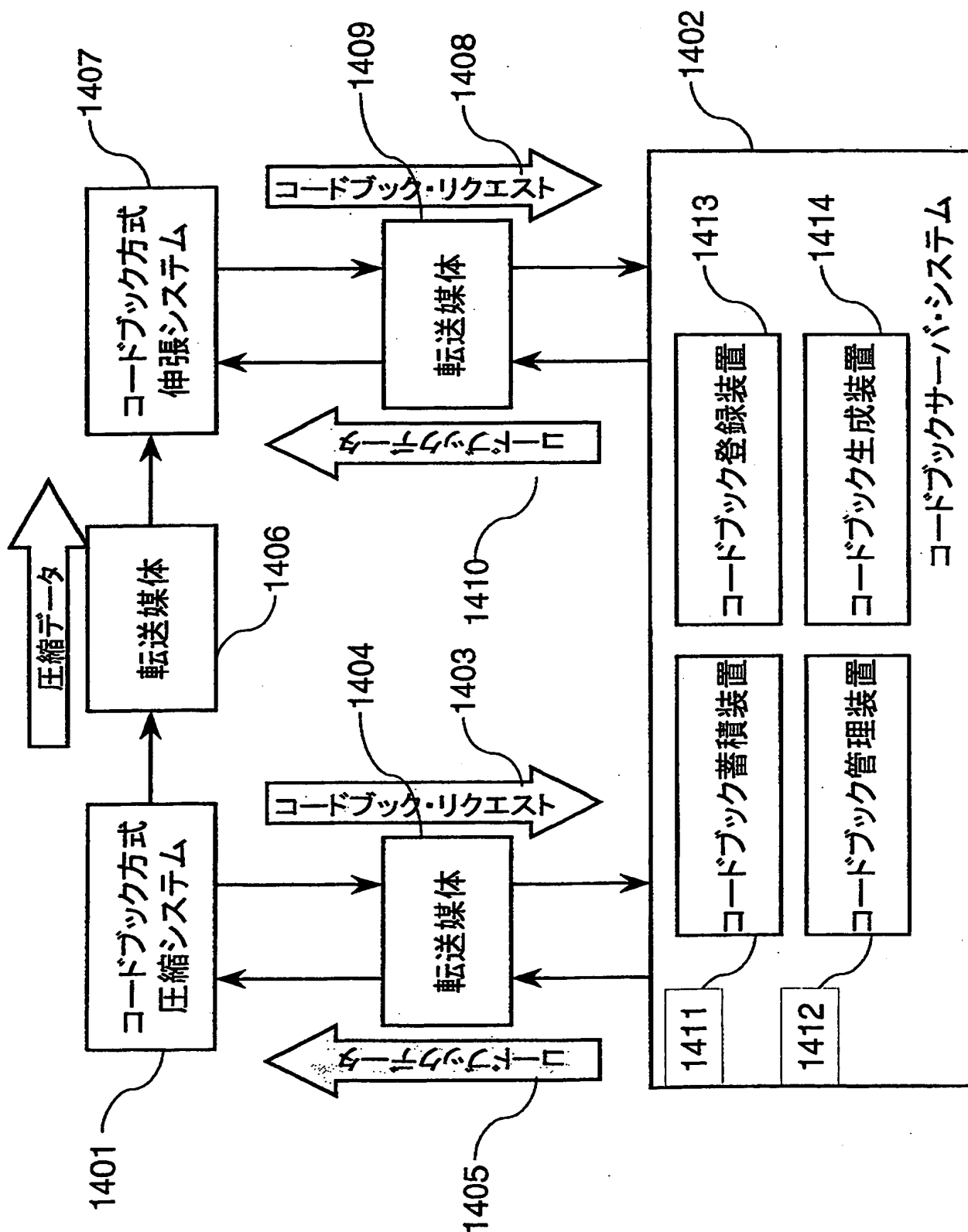


【図 28】





【図 29】



【書類名】 要約書

【要約】

【課題】 本発明は、動画像や音声信号のデータ圧縮を高圧縮率で実現し、圧縮データをもとに質の高いデータを再生しすることを目的とする。

【解決手段】 複数枚の画像データに対する静止画コード番号データ列群の個々の静止画コード番号データ列の同じアドレスに存在するデータを抜きだし、入力された画像の順序に配列された静止画コード番号フレームベクトルを生成し、静止画コード番号フレームベクトル群の個々の静止画コード番号フレームベクトルのデータと時間軸コードブックを構成するベクトル群の個々のデータとの差分をとり絶対値表現したそれぞれの値を加算した結果が最も小さいベクトルが格納されている記憶装置のアドレスの特定を行い、静止画コード番号フレームベクトルに対応するアドレスが持つ時間軸コード番号データとして、記憶装置のアドレス番号を割り当て、時間軸コード番号データ列を生成することを特徴とする。

【選択図】 図8

【書類名】  
【訂正書類】

職権訂正データ  
特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】

000205041

【住所又は居所】

宮城県仙台市青葉区米ヶ袋 2-1-17-301

【氏名又は名称】

大見 忠弘

【特許出願人】

【識別番号】

596089517

【住所又は居所】

東京都文京区本郷 4-1-4

【氏名又は名称】

株式会社ウルトラクリーンテクノロジー開発研究所

【代理人】

申請人

【識別番号】

100088096

【住所又は居所】

東京都千代田区九段南 4-5-11 富士ビル 2 階

【氏名又は名称】

福森 久夫

出 願 人 履 歴 情 報

識別番号 [000205041]

1. 変更年月日 1990年 8月27日

[変更理由] 新規登録

住 所 宮城県仙台市青葉区米ヶ袋2-1-17-301

氏 名 大見 忠弘

出 願 人 履 歴 情 報

識別番号 [596089517]

1. 変更年月日 1996年 6月20日

[変更理由] 新規登録

住 所 東京都文京区本郷4-1-4

氏 名 株式会社ウルトラクリーンテクノロジー開発研究所

